Wireless GPRS-F1/F2 Modems with IP Connectivity

MultiModem® GPRS (MTCBA-G)
ModemModule™ GPRS (MTMMC-G)
SocketModem® GPRS (MTSMC-G)

AT Commands and Application Examples

Reference Guide

MultiTech® Systems

AT Commands for Wireless GPRS-F1/F2 Modems with IP Connectivity
Reference Guide S000333D
Products: MultiModem® GPRS (MTCBA-G-F1/F2)
ModemModule™ GPRS (MTMMC-G-F1/F2)
SocketModem® GPRS (MTSMC-G-F1/F2)

Copyright

This publication may not be reproduced, in whole or in part, without prior expressed written permission from Multi-Tech Systems, Inc. All rights reserved. Copyright © 2003-2007, by Multi-Tech Systems, Inc.
Multi-Tech Systems, Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Multi-Tech Systems, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Multi-Tech Systems, Inc. to notify any person or organization of such revisions or changes.

Trademarks

MultiModem, SocketModem, and the Multi-Tech logo are registered trademarks of Multi-Tech Systems, Inc.
ModemModule is a trademark of Multi-Tech Systems, Inc.

World Headquarters

Multi-Tech Systems, Inc.
2205 Woodale Drive
Mounds View, Minnesota 55112
Phone: 763-785-3500 or 800-328-9717
Fax: 763-785-9874
Internet Address: http://www.multitech.com

Technical Support

Country	By Email	By Phone
France:	support@multitech.fr	+(33) 1-64 61 09 81
Europe, Asia, Africa:	support@multitech.co.uk	+(44) 118 959 7774
U.S., Canada, all others:	support@multitech.com	800-972-2439 or 763-717-5863

Revision History

Revision Level	Date	Description
A	03/18/04	Initial release. **For use with Wavecom IP stack 2.00, firmware version 640b**.
B	05/20/05	**For use with Wavecom IP stack 2.10, firmware version 641c**. Chapters added for UDP and PING. Added AT#RECiADD parameter. Updated text for AT#SMTPUN and AT#SMTPPW. Added Error Codes 35841 (GPRS session lost) and 37123 (IP Connectivity library: Service is running. Unable to set parameter). The phrase IP Connectivity Library replaces SmartStack in the Error Codes. Added the command to erase the flash memory.
C	01/20/06	**For use with Wavecom IP stack 3.02, firmware version 651_**. Added *How to Open TCP/IP Stack* to Chapter 1. To CALLSCREENNUM value Decimal Phone Number, added: *Alphanumeric ASCII text string up to 64 characters*. Removed Value 0 from PPPMODE. Added Response Syntax to DISPLAYIP. Added Syntax section to SENDERADDR. Added new FTP command: FTPMODE. Added FTPMODE and FTPTYPE parameters to FTPGET, FTPPUT, VFTP. New feature in TCP Socket Services is the addition of one more sockets. All TCP commands now include a socket id {1,2}. It affects the TCP Set and Get Syntax. Also, it affects the syntax of: LTCPSTART, LTCPSTOP, OTCP, and VTCP. Added new TCP command: TCPTXDELAY. Added a new syntax example row in the table for OUDP. Added UDPSERV to the possible responses for VUDP. Updated the example table for all commands (VALL) with all the changes above. Added the TCP socket id in the Appendix A examples.
D	05/17/07	Updated the Technical Support contact list and title page.

Contents

Chapter 1 - Introduction **6**

Definitions and Abbreviations **6**

Presentation Rules **6**

Open TCP/IP Stack **6**

Chapter 2 – Dialing Services **7**

Parameter Definitions **7**

#ANSWERMODE Set Answering Mode 7
#CALLBACKTIMER Set Automatic Call-Back Wait 7
#CALLSCREENNUM Set Caller ID Service 7
#REDIALCOUNT Set Connection Attempts Allowed 8
#REDIALDELAY Set Call Retry Delay 8
#PHYTIMEOUT Set Inactivity Time 8
#RINGCOUNT Set Number of Rings/Automatic Operation 8
#DIALN1 Set Primary Dial-Up Number 8
#DIALN2 Set Secondary Dial-Up Number 9
#DIALSELECT Set Internet Connection Number 9
#ISPPW Set ISP Password 9
#ISPUN Set ISP User Name 9
#PPPMODE Set TCP/IP Behavior 9
#PPPMYIP Sets IP Address When Using PPP 10
#PPPPEERIP Sets IP Address for Remote PPP 10
#PPPSERVUN Sets User Name for Remote PPP Client 10
#PPPSERVPW Sets Remote PPP Client Password 10
#APNPW Sets Access Point Name Password 10
#APNSERV Sets Access Point Name 11
#APNUN Sets Access Point User Name 11
#GPRSCID Sets PDP Context Identifier 11
#GPRSMODE Activates Switching Between GSM and GPRS 11

Incoming Call Management **12**

#ACCEPT Answer Incoming Call 12
#CONNECTIONSTOP Stop Communication 12
#CONNECTIONSTART Start Communication 13

Display Parameters **15**

#DISPLAYIP Display IP Addresses 15
#VPPP Display PPP Parameters 15
#VGPRS Display GPRS Parameters 16
#VPHY Display Physical Parameters 16

Chapter 3 – SMTP/POP3 Email Services **17**

Parameter Definitions **17**

#SENDERNAME Set Sender Name 17
#SENDERADDR Set Sender Address 17
#CCREC1 / CCREC2 / CCREC3 Set Additional Recipient CC 17
#DOMAIN Set Domain Name of the Sender 17
#REC1, REC1ADD / REC2, REC2ADD / REC3, REC3ADD Set Email Address 18
#SUBJ1 / SUBJ2 / SUBJ3 Set Email Pre-Defined Subject Text 18
#BODY1 / BODY2 / BODY3 Set Pre-Defined Email Combinations 18
#POP3HEADERMODE Send/Do Not Send POP3 Header 19
#POP3PORT Set POP3 Server 19
#POP3PW Set POP3 Password 19
#POP3SERV Set POP3 Address 19
#POP3UN Set POP3 User Name 19
#SMTPPORT Set SMTP Server Port 20

#SMTPPW Set SMTP Password 20
#SMTPSERV Set SMTP Server Address 20
#SMTPUN Set SMTP User Name 20
#DNSSERV1 Set DNS Server IP Address 20
#DNSSERV2 Set Secondary DNS Server 21
#GETMAIL Retrieve Host Mail 21
#SENDMAIL1 / #SENDMAIL2 / #SENDMAIL3 Send Mail 22
#PUTMAIL Send Host Mail 23
Display Parameters 24
#VMAIL1 / #VMAIL2 / #VMAIL3 Display Email Parameters 24
#VPOP3 Display POP3 Parameters 24
#VSMTP Display SMTP Parameters 24
#VDNS Display DNS Parameters 25

Chapter 4 – FTP Services 26

Parameter Definitions 26
#FTPPORT Enter FTP Server Port 26
#FTPMODE Enter FTP Mode 26
#FTPTYPE Specify Data Type 26
#FTPSERV Specify FTP Server Address 26
#FTPUN Set FTP User Name 27
#FTPPW Set FTP Password 27
#FTPGETFILENAME Set FTP Download File Name 27
#FTPGETPATH Set FTP Path 27
#FTPPUTFILENAME Set FTP Upload File Name 27
#FTPPUTPATH 27
#FTPGET Get Data from Server 28
#FTPPUT Put Data to Server 29
#VFTP Display FTP Parameters 29

Chapter 5 - TCP Socket Services 30

Parameter Definitions 30
#DLEMODE Set DLE Mode 30
#TCPPORT Set TCP Port for Remote Peer 30
#TCPSERV Set Address of Remote TCP Server 30
#LTCPTXDELAY Time Delay 31
#LTCPSTART Open Listening Mode 31
#LTCPSTOP Close Listening Mode 32
#OTCP Open TCP Connection 32
#VTCP Display TCP Parameters 33

Chapter 6 – UDP Socket Services 34

Parameters Definitions 34
#UDPPORT UDP Port Number 34
#UDPSERV UDP Server 34
#UDPTXDELAY UDP Data Delay 34
#LUDPSTART Open an UDP Session in Listen Mode 35
#LUDPSTOP Close a Listening Mode UDP Session 35
#OUDP Open an Active UDP Session 36
#VUDP Display UDP Parameters 36

Chapter 7 – PING Services 37

Parameters Definitions 37
#PINGDELAY PING Delay Time 37
#PINGNUM Number of PING Requests 37
#PINGREMOTE IP Address of PING Request 37
#PING Start PING Request 38
#VPING Display PING Parameters 38

Chapter 8 – Miscellaneous Commands 39

#VVERSION Display Software Version 39
#VSTATE Display Current Status 39
#VALL Display All Parameters 39
#DELFLASH Erase Flash Memory Parameter Values 41

Chapter 9 - Response Messages and Error Codes 42

Response Messages 42
Error Codes 42

Appendix A – Getting Started Examples 44

AT+WOPEN Start TCP/IP Stack 44
AT#VVERSION Get Software Version 44
AT#PPPMODE Set Parameter Value 44
AT+CPIN GSM Network Registration 44
AT+CGREG GSPRS Network Registration State 44
Connect to the Internet 45
Sending/Retrieving Email 46
FTP: Download / Upload Files 47
TCP Socket 48

Appendix B – Detailed Examples 49

Serial Communication 49
Configuring the TCP/IP Stack through the Serial Port 53
Configuring Flow Control over the Serial Port 54
Using the DNS Service 54
Connecting to an Internet Service Provider (ISP) through GSM Data 55
Using the TCP/IP Stack as a PPP Server in GSM Mode 57
Automatic Call-Back to Incoming Calls 58
Automatic Answer of Incoming Calls 60
Answering an Incoming Call 62
Connecting to the Internet through GPRS 64
Retrieving Email 66
Sending an Email Containing Data from the Serial Port 68
Sending a Pre-Defined Email Combination 71
Downloading an FTP File 73
Uploading an FTP File 75
Setting a TCP Port to a Listening Mode 77
Opening a TCP Socket 78

Chapter 1 - Introduction

This document describes the AT parameters and command set available to configure and drive the TCP/IP stack and protocols available on wireless GSM/GPRS products that support IP connectivity.

TCP/IP features and protocols available:

- PPP
- GPRS
- ICMP
- DNS
- SMTP
- POP3
- FTP
- TCP socket

Definitions and Abbreviations

APN – Access Point Name
<DLE> – Escape Character
DNS – Domain Name Server
<ETX> – Escape Character
FTP – File Transfer Protocol
GPRS – General Packet Radio Service
GSM – Global System for Mobile Communication
IP – Internet Protocol
ISP – Internet Service Provider
POP – Post Office Protocol
PPP – Point-to-Point Protocol
PSTN – Public Switched Telephone Network
SMTP – Simple Mail Transfer Protocol
Stack – Low-Level Software
TCP – Transmission Control Protocol

Presentation Rules

In the following chapters, the AT commands are presented with as much precision as possible, through three paragraphs.
Description – The Description paragraph provides general information about the AT command or response behavior.
Values - The Values paragraph provides the parameter values for the AT commands and their corresponding responses.
Syntax – The Syntax paragraph describes the way to use the command, the possible response, or answers.
Schemas are provided where necessary.

Open TCP/IP Stack

Command	Responses
AT+WOPEN=1 *Note: Start the TCP/IP stack*	OK

AT+WOPEN=0
The TCP/IP stack is not opened. In some instances, this may be the default setting.

Chapter 2 – Dialing Services

Parameter Definitions

#ANSWERMODE Set Answering Mode

Definition: The TCP/IP stack can handle incoming calls. This parameter defines how the TCP/IP stack will behave when receiving an incoming call.

Note: The ANSWERMODE parameter must be configured in accordance with ATS0 command, which determines and controls the product automatic answering mode.

Values:

0 **Ignore.** Ignores the incoming call. In this case, it is the responsibility of the host to accept/not accept the incoming call by issuing the AT#ACCEPT command. **Default 0**.

1 **Automatic Answer.** The TCP/IP stack goes off hook and accepts the incoming call. As described below, the calling number must match the one specified in the CALLSCREENNUM parameter. (The RINGCOUNT parameter must be > 0).

2 **Static Callback.** The TCP/IP stack ignores the incoming call and then automatically dials (DIALN1 or DIALN2 determined by the DIALSELECT parameter) by issuing an AT#CONNECTIONSTART command. As described below, the calling number must match the one specified in the CALLSCREENNUM parameter. The RINGCOUNT parameter must be > 0.

3 **Dynamic Callback.** The TCP/IP stack ignores the incoming call and then automatically dials the calling number by issuing an AT#CONNECTIONSTART command. For this feature, the Caller ID service is mandatory. As described below, the calling number must match the one specified in the CALLSCREENNUM parameter. Ensure that the calling phone number is correctly provided by the network.

Syntax: Set Value: AT#ANSWERMODE=<value>

Get Value: AT#ANSWERMODE? or AT#VPHY, AT#VALL

#CALLBACKTIMER Set Automatic Call-Back Wait

Definition: This parameter defines the number of seconds the TCP/IP stack will wait before an automatic call-back operation occurs after receiving an incoming call. It only applies when the ANSWERMODE parameter is set to an automatic call-back mode (value>1). This timer starts at the end of the ringing signal.

Values: Integer between 2 and 255 inclusive. This timer is set in seconds. **Default 2**.

Syntax: Set Value: AT#CALLBACKTIMER=<value>

Get Value: AT#CALLBACKTIMER? or AT#VPHY, AT#VALL

#CALLSCREENNUM Set Caller ID Service

Definition: When receiving an incoming call, the caller identification (Caller ID) service allows the TCP/IP stack to identify the phone number of the remote caller. This information is helpful in preventing unauthorized callers to trigger actions on the TCP/IP stack.

This parameter allows the user to filter the incoming calls when the ANSWERMODE parameter is set to an automatic mode (value>0). This filtering doesn't apply when the ANSWERMODE parameter is set to 0. In this case, it is the host's responsibility to accept or reject the incoming call. If an incoming phone number is not authorized, the TCP/IP stack will ignore it.

Values:

0 Zero. Not remote caller authorized. **Default 0**.

***** All. No filtering is applied on incoming calls. All the remote phone numbers are authorized. This value must be set when wanting to receive incoming calls while the Caller ID service is not available.

Decimal Phone Number. Only the phone number configured here is authorized for incoming calls. Alphanumeric ASCII text string up to 64 characters.

Syntax: Set Value: AT#CALLSCREENNUM=<value>

Get Value: AT#CALLSCREENNUM? or AT#VPHY, AT#VALL

#REDIALCOUNT Set Connection Attempts Allowed

Definition: This parameter indicates how many unsuccessful connection attempts the TCP/IP stack software will make before terminating the connection attempt activity.

Values: Integer between 0 and 14, inclusive. **Default 5.**

If the value is set to 0, the TCP/IP stack software will not make any call retry.

Syntax: Set Value: AT#REDIALCOUNT=<value>

Get Value: AT#REDIALCOUNT? or AT#VPHY, AT#VALL

#REDIALDELAY Set Call Retry Delay

Definition: This parameter controls the delay (in seconds), if any, that will exist between each call retry.

Values: Integer between 5 and 14 inclusive. **Default 5.**

If this parameter is configured to zero, the TCP/IP stack software will attempt another connection immediately after terminating the previous unsuccessful attempt.

Syntax: Set Value: AT#REDIALDELAY=<value>

Get Value: AT#REDIALDELAY? or AT#VPHY, AT#VALL

#PHYTIMEOUT Set Inactivity Time

Definition: This parameter is used by the TCP/IP stack software in order to terminate connections to the telephone line when a long period elapses without activity. "Without activity" is defined as a period when no data is transferred between the Internet and the TCP/IP stack software or between the TCP/IP stack software and the attached equipment. This timer prevents the telephone line from being connected indefinitely for any reason.

Note: When the inactivity timer expires, the modem ends the communication.

Values: Integer between 1 and 255 inclusive. This timer is set in minutes. **Default 15.**

Syntax: Set Value: AT#PHYTIMEOUT=<value**>**

Get Value: AT#PHYTIMEOUT? or AT#VPHY, AT#VALL

Return Codes: TIMEOUT: The inactivity timer is reached; the modem ends the communication.

#RINGCOUNT Set Number of Rings/Automatic Operation

Definition: This parameter defines the number of rings that will be counted before an automatic operation occurs when receiving an incoming call.

It applies only when the ANSWERMODE parameter is set to an automatic mode (value>0).

If the ANSWERMODE parameter is used (value not 0), the RINGCOUNT value must also be >0.

Note: The RINGCOUNT parameter must be configured in accordance with the ATS0 configuration to prevent conflicting interaction with the two features.

Values: Integer between 0 and 15 inclusive. **Default 0.**

Syntax: Set Value: AT#RINGCOUNT=<value>

Get Value: AT#RINGCOUNT? or AT#VPHY, AT#VALL

#DIALN1 Set Primary Dial-Up Number

Definition: This parameter is the primary dial-up phone number that will connect with the local ISP. Length depends on country/region.

Values: Decimal phone numbers. **No default.**

Syntax: Set Value: AT#DIALN1=<value>

Get Value: AT#DIALN1? or AT#VPHY, AT#VALL

#DIALN2 Set Secondary Dial-Up Number

Definition: This parameter is the secondary dial-up phone number that will connect with the local ISP. Length depends on country.

Values: Decimal phone numbers. **No default.**

Syntax: Set Value: AT#DIALN2=<value>

Get Value: AT#DIALN2? or AT#VPHY, AT#VALL

#DIALSELECT Set Internet Connection Number

Definition: The value of this command determines the number called to establish an Internet connection. It configures the TCP/IP stack software to use the primary dial-up number or the secondary dial-up number.

Values:

1 Use primary dial-up number. **Default 1.**

2 Use secondary dial-up number

Syntax: Set Value: AT#DIALSELECT=<value>

Get Value: AT#DIALSELECT? or AT#VPHY, AT#VALL

#ISPPW Set ISP Password

Definition: This parameter sets the password for the ISP account. When communication is initiated and once the physical (modem) connection has been established with the ISP, the TCP/IP stack software must provide the ISP with the password associated with the account to be used.

Values: Alpha-numeric ASCII text string up to 64 characters. **No default.**

Syntax: Set Value: AT#ISPPW=<value>

Get Value: AT#ISPPW? or AT#VPPP, AT#VALL

#ISPUN Set ISP User Name

Definition: This parameter sets the user name of the ISP account. When communication is initiated and the physical (modem) connection has been established with the ISP, the TCP/IP stack software must provide the ISP with the user name associated with the account to be used.

Values: Alpha-numeric ASCII text string up to 64 characters. **No default.**

Syntax: Set Value: AT#ISPUN=<value>

Get Value: AT#ISPUN? or AT#VPPP, AT#VALL

#PPPMODE Set TCP/IP Behavior

Definition: The TCP/IP stack can manage the access layer in different ways. This parameter selects the way the TCP/IP stack must run once the physical is layer successfully established.

Values: Integer between 0 and 4 inclusive.

1 **Standard PPP.** The TCP/IP stack behaves as a PPP client for outgoing calls and as a PPP server for incoming calls. **Default 1.**

2 **Reverse PPP.** The TCP/IP stack behaves as a PPP server for outgoing calls and as a PPP client for incoming calls.

3 **PPP client only.** The TCP/IP stack always behaves as a PPP client for both outgoing and incoming calls.

4 **PPP server only.** The TCP/IP stack always behaves as a PPP server for both outgoing and incoming calls.

9 **Specific.** This case is reserved for specific behavior.

Syntax: Set Value: AT#PPPMODE=<value>

Get Value: AT#PPPMODE? or AT#VPPP, AT#VALL

#PPPMYIP Sets IP Address When Using PPP

Definition: When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter setting), it is in charge of the IP address attribution mechanism.

Once the PPP authentication is successfully achieved, the remote PPP peer asks the TCP/IP stack for an IP address. Then the related PPP layer, called IPCP, suggests an IP address to the peer that has been previously stored in the TCP/IP stack parameters. If the remote accepts this address, the IP link is then established.

This parameter defines the IP address to be attributed to the TCP/IP stack when the PPP Server mode is running.

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx). **Default 0.0.0.0**

Syntax: Set Value: AT#PPPMYIP=<value>

Get Value: AT#PPPMYIP? or AT#VPPP, AT#VALL

#PPPPEERIP Sets IP Address for Remote PPP

Definition: When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter setting), it is in charge of the IP address attribution mechanism.

Once the PPP authentication is successfully achieved, the remote PPP peer asks the TCP/IP stack for an IP address. Then the related PPP layer, called IPCP, suggests an IP address to the peer that has been previously stored in the TCP/IP stack parameters. If the remote accepts this address, the IP link is then established.

This parameter defines the IP address to be attributed to the remote PPP peer when the PPP Server mode is running.

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx). **Default 0.0.0.0**

Syntax: Set Value: AT#PPPPEERIP=<value>

Get Value: AT#PPPPEERIP? or AT#VPPP, AT#VALL

#PPPSERVUN Sets User Name for Remote PPP Client

Definition: When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter), it checks the remote PPP client login/password before granting access to the server.

This parameter defines the user name that must be specified by the remote PPP client.

Values: Alpha-numeric ASCII text string up to 64 characters. **No default.**

Syntax: Set Value: AT#PPPSERVUN=<value>

Get Value: AT#PPPSERVUN? or AT#VPPP, AT#VALL

#PPPSERVPW Sets Remote PPP Client Password

Definition: When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter setting), it checks the remote PPP client login/password before granting access to the server.

This parameter defines the password that must be specified by the remote PPP client.

Values: Alpha-numeric ASCII text string up to 64 characters. **No default.**

Syntax: Set Value: AT#PPPSERVPW=<value>

Get Value: AT#PPPSERVPW? or AT#VPPP, AT#VALL

#APNPW Sets Access Point Name Password

Definition: #APNPW is the Access Point Name password parameter coming with the APNUN from the GSM operator and provides GPRS access.

Values: Alphanumeric ASCII text string up to 120 characters. **No default.**

Syntax: Set Value: AT#APNPW=<value>

Get Value: AT#APNPW? or AT#VGPRS, AT#VALL

#APNSERV Sets Access Point Name

Definition: #APNSERV is the Access Point Name parameter coming from the GSM operator for providing GPRS access.

Values: Alphanumeric ASCII text string up to 120 characters. **No default.**

Syntax: Set Value: AT#APNSERV=<value>

Get Value: AT#APNSERV? or AT#VGPRS, AT#VALL

#APNUN Sets Access Point User Name

Definition: #APNUN is the Access Point Name User Name parameter coming with the APNPW from the GSM operator for providing GPRS access.

Values: Alphanumeric ASCII text string up to 120 characters. **No default.**

Syntax: Set Value: AT#APNUN=<value>

Get Value: AT#APNUN? or AT#VGPRS, AT#VALL

#GPRSCID Sets PDP Context Identifier

Definition: #GPRSCID is the PDP context identifier which specifies a particular PDP context definition. This parameter is local and may be used in other PDP context-related commands.

Values: Numeric between 1 and 4 inclusive. **Default 1.**

Syntax: Set Value: AT#GPRSCID=<value>

Get Value: AT#GPRSCID? or AT#VGPRS, AT#VALL

#GPRSMODE Activates Switching Between GSM and GPRS

Definition: #GPRSMODE configures the activation of the software for switching between GSM and GPRS.

Values: **0** Configured for GSM use.

1 Configured for GPRS use. **Default 1**.

Syntax: Set Value: AT#GPRSMODE=<value>

Get Value: AT#GPRSMODE? or AT#VGPRS, AT#VALL

Incoming Call Management

#ACCEPT Answer Incoming Call

Definition: This command directs the TCP/IP stack to answer an incoming call. When the TCP/IP stack receives an incoming call, it sends over the serial port the "RING" messages. Depending on the value of the ANSWERMODE parameter, the TCP/IP stack may answer automatically or not.

If ANSWERMODE is set to 0, it is the host that is responsible for answering the incoming call.

Once the physical layer is up, the TCP/IP stack runs the applicable protocol as specified in the PPPMODE parameter.

The AT#CONNECTIONSTOP command ends the connection by going on hook.

Values: **Read parameters:**

ANSWERMODE
IPSPW
ISPUN
PPPMODE
PPPMYIP
PPPPEER (if PPPMODE is set in server mode)
PPPPEERIP
PPPSENDPW
PPPSENDUN

Syntax: **AT#ACCEPT**

Command	Possible Responses
AT#ACCEPT Note: Manual acceptance of an incoming call	OK Note: Beginning of the call setting process CONNECT <speed> Note: Modem speed negotiated between both sides xxx.xxx.xxx.xxx Note: IP address indication attributed to the TCP/IP stack PPP OK Note: The software is ready to run IP applications to send/receive data
AT#ACCEPT Note: Manual acceptance of an incoming call	NO CARRIER Note: The modem handshaking process with the remote host is interrupted or unsuccessful
AT#ACCEPT Note: Manual acceptance of an incoming call	PPP ERROR Note: The PPP negotiation has failed (check ISPUN, ISPPW, PPPMODE, and the configuration of the PPP peer)

#CONNECTIONSTOP Stop Communication

Definition: This command directs the TCP/IP stack to end a GPRS or GSM communication previously established with a START command or AT#ACCEPT.

Values: No TCP/IP parameter is used for the execution of this command.

Syntax: **AT#CONNECTIONSTOP**

Command	Possible Responses
AT#CONNECTIONSTOP Note: Disconnect	OK Note: Phone line is released

#CONNECTIONSTART Start Communication

Definition: This command directs the TCP/IP stack to originate an outgoing call.

Upon receiving this command, the TCP/IP stack attempts to complete a connection session according to the GPRSMODE parameter, selecting the GSM or GPRS mode.

- In GSM mode, the TCP/IP stack will dial the number according to the Dial Option parameter (DIALN1 or DIALN2 depending on DIALSELECT). If an error occurs, the TCP/IP stack automatically re-attempts the call origination attempt, according to the REDIALCOUNT parameter. Once the physical layer is up, the TCP/IP stack runs the applicable protocol as specified in the PPPMODE parameter.
- In GPRS mode, the TCP/IP stack will establish a GPRS session with the APN using APNUN, APNPW, GPRSCID parameters. Once the GPRS link is up, the product is connected to the Internet. The AT#CONNECTIONSTOP command closes the GSM or GPRS connection by going on hook.

Values:

Read Parameters for GSM Mode:

DIALN1
DIALN2
DIALSELECT
ISPPW
ISPUN
REDIALCOUNT
REDIALDELAY

If PPPMODE set to 2 or 4:

PPPMYIP
PPPPEERIP
PPPSERVPW
PPPSERVUN

For GPRS Mode:

APNPW
APNSERV
APNUN
GPRSCID

Syntax: **AT#CONNECTIONSTART**

Command (GSM Mode)	Possible Responses
AT#CONNECTIONSTART Note: Request connection to GSM network	DIALING Note: Phone line is available 2124560123 Note: Dial DIALN1 or DIALN2 number depending on DIALSELECT CONNECT 9600 Note: Modem speed negotiated between both sides. If the TCP/IP stack is configured for modem only operation (PPPMODE parameter), there are no more return codes 213.192.200.4 Note: IP address attributed to the TCP/IP stack OK_Info_PPP Note: As soon as the TCP/IP stack software displays this message, it is ready to receive commands
AT#CONNECTIONSTART	BUSY Note: A busy signal is detected on the remote site TCP/IP stack will wait REDIALDELAY seconds then dial again. The redialing will continue until success or until the number of call retries defined in parameter REDIALCOUNT has been reached.

GSM Mode for AT#CONNECTIONSTART - continued

Command (GSM Mode)	Possible Responses
AT#CONNECTIONSTART	NO ANSWER *Note: There is no response from the remote site.* *TCP/IP stack will wait REDIALDELAY seconds then dial again.* *The redialing will continue until success or until the number of call retries defined in parameter REDIALCOUNT has been reached.*
AT#CONNECTIONSTART	NO CARRIER *Note: The modem handshaking process with the remote host is interrupted or unsuccessful.* *TCP/IP stack will wait REDIALDELAY seconds then dial again.* *The redialing will continue until success or until the number of call retries defined in parameter REDIALCOUNT has been reached.*
AT#CONNECTIONSTART	#CME ERROR : 37120 #CME ERROR : 37121 #CME ERROR : 28980 #CME ERROR : 28981 *Note: (only one response at a time): The PPP negotiation has failed (check ISPUN, ISPPW, PPPMODE)*
AT#CONNECTIONSTART	#CME ERROR: 35865 *Note: The product is not registered on the network*
AT#CONNECTIONSTART	#CME ERROR: 35840 *Note: The product is already running (host is connected)*

GPRS Mode

Command (GPRS Mode)	Possible Responses
AT#CONNECTIONSTART Note: Request connection to GPRS network	213.192.200.4 *Note: IP address attributed to the TCP/IP stack* OK_Info_GPRSActivation *Note: GPRS session established and product connected to the Internet*
AT#CONNECTIONSTART	#CME ERROR: 35866 *Note: All connection attempts will return this message if the GPRS session cannot be established*
AT#CONNECTIONSTART	#CME ERROR: 35865 *Note: The product is not registered on the network*
AT#CONNECTIONSTART	#CME ERROR: 35868 *Note: Aborted GPRS connection; check APN parameters*

Display Parameters

#DISPLAYIP Display IP Addresses

Definition: This command allows the attached host to view the attributed IP addresses that have been attributed during the IPCP phase of the PPP negotiation. Both local and remote PPP peer IP addresses are displayed. This command should be issued only once the PPP OK message has been received from the TCP/IP stack.

Values: **Read Parameters:**
PPPMYIP
PPPPEERIP

Syntax: **AT#DISPLAYIP**

Response Syntax: **MY IP: xxx.xxx.xxx.xxx**
PEER IP: xxx.xxx.xxx.xxx

GSM Mode

Command (GSM Mode)	Possible Responses
AT#DISPLAYIP *Note: Request for local and remote IP addresses*	MY IP: 1.2.3.4 PEER IP: 1.0.3.5 OK *Note: PPPMYIP (IP address attributed to the TCP/IP stack) and PPPPEERIP (IP address attributed to the PPP peer) parameter values*
AT#DISPLAYIP *Note: Request for local and remote IP addresses*	#CME ERROR: 35867 *Note: No IP addresses have been attributed: no active connection or PPP/IPCP negotiation not yet completed*

GPRS Mode

Command (GPRS Mode)	Possible Responses
AT#DISPLAYIP *Note: Request for local and remote IP addresses*	MY IP: 1.2.3.4 Gateway IP: 1.0.3.5 OK *Note: PPPMYIP (IP address attributed to the TCP/IP stack) and PPPPEERIP (IP address attributed to the PPP peer) parameter values*
AT#DISPLAYIP *Note: Request for local and remote IP addresses*	#CME ERROR: 35867 *Note: No IP addresses have been attributed: no active connection or PPP/IPCP negotiation not yet completed*

#VPPP Display PPP Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the PPP layer configuration.

Values: **Read Parameters:**
ISPPW
ISPUN
PPPMODE
PPPMYIP
PPPPEERIP
PPPSERVUN
PPPSERVPW

Syntax: **AT#VPPP**

Command	Possible Responses
AT#VPPP	#ISPUN: “myispun” #ISPPW: “myisppwd: #PPPMODE: 1 #PPPMYIP: “0.0.0.0” #PPPPEERIP: “0.0.0.0” #PPPSERVUN: “myname” #PPPSERVPW: “mypasswd” OK

#VGPRS Display GPRS Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the GPRS configuration.

Values:
APNPW
APNSERV
APNUN
GPRSCID

Syntax: **AT#VGPRS**

Command	Possible Responses
AT#VGPRS	#GPRSCID: 1 #APNPW: “access” #APNSERV: “a2myoperator.com” #APNUN: “a2b” OK

#VPHY Display Physical Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the physical layer configuration.

Values:
ANSWERMODE
CALLBACKTIMER
CALLSCREENNUM
DIALN1
DIALN2
DIALSELECT
GPRSMODE
PHYTIMEOUT
REDIALCOUNT
REDIALDELAY
RINGCOUNT

Syntax: **AT#VPHY**

Command	Possible Responses
AT#VPHY	#ANSWERMODE: 0 #CALLBACKTIMER: 2 #CALLSCREENNUM: “0” #DIALN1: “” #DIALN2: “” #DIALSELECT: 1 #GPRSMODE: 1 #PHYTIMEOUT: 15 #REDIALCOUNT: 5 #REDIALDELAY: 5 #RINGCOUNT: 0 OK

Chapter 3 – SMTP/POP3 Email Services

Parameter Definitions

#SENDERNAME Set Sender Name

Definition: The sender's literal name (different from the SENDERADDR parameter, which is the sender's email address). This parameter will appear in the header of the email sent by the TCP/IP stack software, in the field: "From:"

Values: Alphanumeric ASCII text string up to 120 characters. The address must be provided in literal format (for instance, "machine 245"). **No default.**

Syntax: Set Value: AT#SENDERNAME=<value>

Get Value: AT#SENDERNAME? or AT#VSMTP, AT#VALL

#SENDERADDR Set Sender Address

Definition: To send emails, the TCP/IP stack software must know the email address of the sender. The "sender" is the email identification of the hardware platform itself or the optional attached equipment. This email address will appear in the header of the email sent by the TCP/IP stack software, in the field "From:"

Values: Alphanumeric ASCII text string up to 120 characters. The address must be provided in literal format (for instance xxxxxxx@web.zyx). **No default.**

Syntax: Set Value: AT#SENDERADDR=<value>

Get Value: AT#SENDERADDR? or AT#VSMTP, AT#VALL

#CCREC1 / CCREC2 / CCREC3 Set Additional Recipient CC

Definition: The software can send email messages to an additional recipient as a "carbon copy". This parameter contains the email address of the additional recipient. This email address will appear in the header of the email sent by the TCP/IP stack software in the field "Cc:"

For a given value *n*, the CCRECn parameter is directly associated with the RECn parameter.

Values: Alphanumeric ASCII text string up to 120 characters. The address must be provided in literal format (for instance xxxxx@web.com). **No default.**

Syntax: Set Value: AT#CCRECi=<value> (replace i with 1, 2, or 3)

Get Value: AT#CCRECi? (replace i with 1, 2, or 3) or AT#VMAILi, AT#VALL

#DOMAIN Set Domain Name of the Sender

Definition: When sending an email message, the TCP/IP stack software must provide the SMTP server with the domain name of the sender. In some cases, this domain name may be different from the domain name included in the sender's email address.

Values: Alphanumeric ASCII text string up to 120 characters. **No default.**

Syntax: Set Value: AT#DOMAIN=<value>

Get Value: AT#DOMAIN? or AT#VSMTP, AT#VALL

#REC1, REC1ADD / REC2, REC2ADD / REC3, REC3ADD Set Email Address

Definition: To send email messages, the TCP/IP stack software must know the e-mail address of the recipient. This email address will appear in the email header sent by the TCP/IP stack software, in the field 'To:'

The RECi parameter can hold a maximum of 10 email addresses, each email address being at the most 120 characters long.

Values: RECi (i = 1, 2, or 3)

Alphanumeric ASCII text string up to 120 characters. The address must be provided in literal format (for instance xxxxx@company.com). **No default.**

Setting One Email Address / Resetting the Parameter / Getting:

Set value / reset the parameter: AT#RECi="Value" (replace i with 1, 2, or 3)

Get value: AT#REC? (replace i with 1, 2, or 3) or AT#VMAILi, AT#VALL

Setting One to Ten Email Address to the RECi Parameter / Resetting the Parameter:

To set 1 to 10 email addresses to the RECi parameter, enter the AT#RECiADD<CR> **overwriting** command.

Each email address has to be an alphanumeric ASCii text string, in literal format (for instance, dev12345678@web.xyz). To add another email address, enter the <CRLF> pair. To end the setting, enter the following character: 1A (in ASCII code), generated in a keyboard by CTRL+Z escape sequence.

Note1: This command overwrites all previously set email addresses of the RECi parameter.

Note2: If more than ten addresses are entered, the 11th and subsequent addresses until the end character will be ignored.

Example: AT#REC1ADD<CR>

Email1@domain.fr<CRLF>

Email2@domain.com<CRLF>

<CTRL+Z>

#SUBJ1 / SUBJ2 / SUBJ3 Set Email Pre-Defined Subject Text

Definition: These parameters contain the pre-defined subject text that will be used by the TCP/IP stack to compose the email header.

Values: Alphanumeric ASCII text string up to 120 characters. **No default.**

Syntax: Set Value: AT#SUBJi=<value> (replace i with 1, 2, or 3)

Get Value: AT#SUBJi? (replace i with 1, 2, or 3) or AT#VMAILi, AT#VALL

#BODY1 / BODY2 / BODY3 Set Pre-Defined Email Combinations

Definition: These parameters store pre-defined message bodies. They allow the host application to send pre-defined email combinations.

Values: The body content has to be entered after the AT#BODY1<CR> command. It has to be an alphanumeric ASCII text string up to 120 characters followed this character: 1A (in ASCII code) and generated on a keyboard by CTRL+Z. **No default.**

Syntax: Set Value: AT#BODYi=<value> (replace i with 1, 2, or 3)

Get Value: AT#BODYi? (replace i with 1, 2, or 3) or AT#VMAILi, AT#VALL

Example: AT#BODY1<CR>

Text string

<CTRL+Z>

#POP3HEADERMODE Send/Do Not Send POP3 Header

Definition: When receiving an email message, the TCP/IP stack can be configured to send or not to send the POP3 header over the serial port. The POP3 header contains the From, Cc and Subject fields.

Values:
0 The email header will not be sent over the serial port while retrieving.
1 The email header will be sent over the serial port while retrieving. **Default.**

Syntax: Set Value: AT#POP3HEADERMODE=<value>
Get Value: AT#POP3HEADERMODE? or AT#VPOP3, AT#VALL

#POP3PORT Set POP3 Server

Definition: To reach the POP3 server, the TCP/IP stack software must know the port of the POP3 server used for retrieving email.

Note: This parameter should be changed only by your network administrator since it depends on network infrastructure configuration including firewalls, proxy or specific TCP port translation settings.

Values: 5 digits (each digit between 0 and 9 inclusive). Note that numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header. **Default 110.**

Syntax: Set Value: AT#POP3PORT=<value>
Get Value: AT#POP3PPORT? or AT#VPOP3, AT#VALL

#POP3PW Set POP3 Password

Definition: Password for POP3 account. To retrieve email messages sent to a specified email address, the TCP/IP stack software must know the POP3 password that has been set for that email account.

Values: Alphanumeric ASCII text string up to 64 characters. **No default.**

Syntax: Set Value: AT#POP3PW=<value>
Get Value: AT#POP3PW? or AT#VPOP3, AT#VALL

#POP3SERV Set POP3 Address

Definition: To retrieve email messages, the TCP/IP stack software must know the address of the POP3 server that is to be used. The POP3 server must be the one where the specified email account is hosted (which is not necessarily maintained by the local ISP).

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or
Alphanumeric ASCII text string up to 120 characters if DNS is available. **No default.**

Syntax: Set Value: AT#POP3SERV=<value>
Get Value: AT#POP3SERV? or AT#VPOP3, AT#VALL

#POP3UN Set POP3 User Name

Definition: User name for POP3 account. To retrieve email messages sent to a specified email address, the TCP/IP stack software must know the POP3 user name that has been set for that email account.

Values: Alphanumeric ASCII text string up to 64 characters. **No default.**

Syntax: Set Value: AT#POP3UN=<value>
Get Value: AT#POP3UN? or AT#VPOP3, AT#VALL

#SMTPPORT Set SMTP Server Port

Definition: To reach the SMTP server, the TCP/IP stack software must know the port of the SMTP server used for sending email.

Note: This parameter should be changed only by your network administrator since it depends on network infrastructure configuration including firewalls, proxy or specific TCP port translation settings.

Values: From 1 to 5 digits (each digit between 0 and 9 inclusive). Note that numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header. **Default 25.**

Syntax: Set Value: AT#SMTPPORT=<value>

Get Value: AT#SMTPPORT? or AT#VSMTP, AT#VALL

#SMTPPW Set SMTP Password

Definition: To send email messages, some SMTP servers are using an authentication process. In these cases, the TCP/IP stack software will provide the SMTP password (associated to the SMTP user name) for the email sending process.

If this parameter is an empty string, the authentication mode is inactive.

If both this parameter and the SMTPUN parameter are not empty, the authentication mode is active.

Values: Alphanumeric ASCII text string up to 64 characters. **No default.**

Syntax: Set Value: AT#SMTPPW=<value>

Get Value: AT#SMTPPW? or AT#VSMTP, AT#VALL

#SMTPSERV Set SMTP Server Address

Definition: To send email messages the TCP/IP stack software must know the address of the SMTP server that is to be used. In most cases, the local ISP maintains the SMTP server.

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or

Alphanumeric ASCII text string up to 120 characters if DNS is available. **No default.**

Syntax: **Set Value: AT#SMTPSERV=<value>**

Get Value: AT#SMTPSERV? or AT#VSMTP, AT#VALL

#SMTPUN Set SMTP User Name

Definition: To send email messages, some SMTP servers use an authentication process. In these cases, the TCP/IP stack software will provide the SMTP user name (associated with a SMTP password) for the email sending process.

If this parameter is an empty string, the authentication mode is inactive.

If both this parameter and the SMTPPW parameter are not empty, the authentication mode is active.

Values: Alphanumeric ASCII text string up to 64 characters. **No default.**

Syntax: Set Value: AT#SMTPUN=<value>

Get Value: AT#SMTPUN? or AT#VSMTP, AT#VALL

#DNSSERV1 Set DNS Server IP Address

Definition: In order to translate the server names from literal format into IP addresses, the TCP/IP stack software implements the Domain Name System (DNS) protocol. The DNS Server IP address must be specified for use by the TCP/IP stack software.

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx). **Default 0.0.0.0**

Syntax: Set Value: AT#DNSSERV1=<value>

Get Value: AT#DNSSERV1? or AT#VDNS, AT#VALL

#DNSSERV2 Set Secondary DNS Server

Definition: In order to translate the server names from literal format into IP addresses, the TCP/IP stack software implements the Domain Name System (DNS) protocol. The DNS Server IP address has to be specified for use by the TCP/IP stack software. This secondary DNS server is used in the case where the primary DNS server does not respond to a request.

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx). **Default 0.0.0.0**

Syntax: Set Value: AT#DNSSERV2=<value>

Get Value: AT#DNSSERV2? or AT#VDNS, AT#VALL

#GETMAIL Retrieve Host Mail

Definition: This command allows the attached host to direct the TCP/IP stack to retrieve the first mail present in the POP3 server list. Once an IP link is established, the attached host can retrieve an email message at any time (except when the TCP/IP stack software is already in a process using TCP resources). This command is similar to a “check email box” feature issued by a standard messaging client on a PC.

Values: **Read Parameters:**
POP3HEADERMODE
POP3PORT
POP3UN
POP3PW
POP3SERV

Syntax: **AT#GETMAIL**

Command	Possible Responses
AT#GETMAIL Note: Retrieve mail	OK_Info_Mail <mail content> **Notes:** This message is issued when one email message is located in the specified POP3 mailbox. Depending on the POP3HEADERMODE parameter, the TCP/IP stack sends the email header over the serial port to the attached host. The (CR)(LF)(.)(CR)(LF) sequence finally indicates the end of the email body.
AT#GETMAIL	OK_Info_NoMail **Note:** There is no email to retrieve in the POP3 mailbox.
AT#GETMAIL	#CME ERROR: 38027 **Notes:** The address of the POP3 server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong POP3 server address has been filled in.
AT#GETMAIL	#CME ERROR: <value> **Notes:** An error has occurred during the communication with the remote POP3 server. It may also happen during the data transfer after the MAIL message. In this case, it is preceded by a (CR)(LF)(.)(CR)(LF) sequence. This error can be due to one of the following reasons: - The DNS servers are not able to resolve the POP3 server address - The POP3 server is temporarily out of service - The authentication (POP3UN, POP3PW) is not valid See section ‘ Response messages and error codes’.

#SENDMAIL1 / #SENDMAIL2 / #SENDMAIL3 Send Mail

Definition: This command sends one of the 3 pre-defined email combinations. Once an IP link is established, the attached host can direct the TCP/IP stack to send an email message at any time (except when the TCP/IP stack software is already in a process using TCP resources).

The header of this email is built using the REC1/2/3, CCREC1/2/3 and SUBJ1/2/3 parameters while the body is filled in the BODY1/2/3 parameter.

This command is similar to a "send email" operation issued by a standard messaging client on a PC.

Note: #SENDMAILi is used for #SENDMAIL1, #SENDMAIL2, or #SENDMAIL3.

Values: **Read Parameters:**
REC1 or REC2 or REC3
CCREC1 or CCREC2 or CCREC3
SUBJ1 or SUBJ2 or SUBJ3
BODY1 or BODY2 or BODY3
SENDERADDR
SENDERNAME
DOMAIN
SMTPPORT
SMTPSERV
SMTPPW
SMTPUN

Syntax: **AT#SENDMAILi**

Command	Possible Responses
AT#SENDMAIL1 Note: Send predefined mail #1	OK **Note:** Mail 1 has been successfully sent
AT#SENDMAIL2 Note: Send predefined mail #2	OK Note: Mail 2 has been successfully sent
AT#SENDMAIL3 Note: Send predefined mail #3	OK **Note:** Mail 3 has been successfully sent
AT#SENDMAIL2	#CME ERROR: 38027 **Notes:** The address of the SMTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong SMTP server address has entered.
AT#SENDMAIL1	#CME ERROR: <value> **Notes:** An error has occurred during the communication with the remote SMTP server. It may also happen during the data transfer (after the OK message). This error can be due to one of the following reasons: - The DNS servers are not able to resolve the SMTP server address - The SMTP server is temporarily out of service - The authentication (SMTPUN, SMTPPW) is not valid - An email address specified in REC1 or CCREC1 is not valid See section 'Response messages and error codes'.

#PUTMAIL Send Host Mail

Definition: This command allows the attached host to send an email message containing body text passed to the TCP/IP stack over the serial port. Once an IP link is established, the attached host can send an email message at any time (except when the TCP/IP stack software is already in a process using TCP resources).

The header of this email is built using the REC1, CCREC1 and SUBJ1 parameters.

This command is similar to a "send email" operation issued by a standard messaging client on a PC.

Values: **Read Parameters:**
REC1
CCREC1
SUBJ1
SENDERADDR
SENDERNAME
DOMAIN
SMTPPORT
SMTPSERV
SMTPPW
SMTPUN

Syntax: **AT#PUTMAIL**

Command	Possible Responses
AT#PUTMAIL Note: You have to configure only receiver address1, copy address1, and subject1 before or during the session, but content (body) of the email is typed when the TCP/IP session is established. Content is not echoed.	OK_Info_WaitingForData **Notes:** An SMTP session has been successfully opened with the remote SMTP server. Once the TCP/IP stack has displayed this message, all the data received on the serial port is sent within the email body. The (CR)(LF)(.)(CR)(LF) sequence sent by the attached host indicates the TCP/IP stack the end of the email body.
<content> Note: Content is not written when typing	
<CR><LF> . <CR><LF> Note: Termination sequence <CR> = <Enter>, <LF> = <Ctrl Enter>	OK **Note:** The mail has been successfully sent.
AT#PUTMAIL	#CME ERROR: 38027 **Notes:** The address of the SMTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong SMTP server address has been filled in.
AT#PUTMAIL	#CME ERROR: <value> **Notes:** An error has occurred during the communication with the remote SMTP server. It may also happen during the data transfer (after the OK message). This error can be due to one of the following reasons: - DNS servers are not able to resolve the SMTP server address. - SMTP server is temporarily out of service. - Authentication (SMTPUN, SMTPPW) is not valid. - An email address specified in REC1 or CCREC1 is not valid. - An inactivity period of 50 seconds on the serial port. See section 'Response messages and error codes'

Display Parameters

#VMAIL1 / #VMAIL2 / #VMAIL3 Display Email Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the email combinations configuration.

Notes:

- The BODY1 / BODY2 / BODY3 parameter values are not displayed. Their values can be displayed using the AT#BODY1? / AT#BODY2? / AT#BODY3? Commands.
- #VMAILi is used for #VMAIL1 or #VMAIL2 or #VMAIL3.

Values: **Read Parameters:**
RECi (REC1 or REC2 or REC3)
CCRECi (CCREC1 or CCREC2 or CCREC3)
SUBJi (SUBJ1 or SUBJ2 or SUBJ3)

Syntax: **AT#VMAILi**

Command	Possible Responses
AT#VMAIL1 Note: View predefined (nb 1) mail header elements	#REC1= #CCREC1= #SUBJ1= OK

#VPOP3 Display POP3 Parameters

Definition: Directs the TCP/IP stack to display all the AT# parameters related to the email retriever configuration.

Values: **Read Parameters:**
POP3HEADERMODE
POP3PORT
POP3PW
POP3SERV
POP3UN

Syntax: **AT#VPOP3**

Command	Possible Responses
AT#VPOP3 Note: View POP3 parameters	#POP3HEADERMODE: 1 #POP3PORT: 110 #POP3PW: “mypop3passwd” #POP3SERV: “pop3server” #POP3UN: “mypop3un” OK

#VSMTP Display SMTP Parameters

Definition: Directs the TCP/IP stack to display all the AT# parameters related to the email sender configuration.

Values: **Read Parameters:**
DOMAIN
SENDERADDR
SENDERNAME
SMTPPORT
SMTPPW
SMTPSERV
SMTPUN

Syntax: **AT#VSMTP**

Command	Possible Responses
AT#VSMTP Note: View SMTP parameters	#DOMAIN: “a2myoper.com” #SENDERADDR: “toto@myoper.com” #SENDERNAME: “toto” #SMTPPORT: 25 #SMTPPW: “mysmtppw” #SMTPSERV: “smtp.a2myoper.com” #SMTPUN: “mysmtpun” OK

#VDNS Display DNS Parameters

Definition: Directs the TCP/IP stack to display all the AT# parameters related to the DNS servers configuration.

Values: **Read Parameters:**
DNSSERV1
DNSSERV2

Syntax: **AT#VDNS**

Command	Possible Responses
AT#VDNS	#DNSSERV1: “0.0.0.0” #DNSSERV2: “0.0.0.0” OK

Chapter 4 – FTP Services

Parameter Definitions

#FTPPORT Enter FTP Server Port

Definition: To reach the FTP server, the TCP/IP stack software must know the control port of the FTP server used for file transfer.

Note: This parameter should be changed only upon request of your network administrator since it applies to network infrastructure including firewalls, proxy or specific TCP port translation.

Values: From 1 to 5 digits (each digit between 0 and 9 inclusive).

Note that numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header. **Default 21**.

Syntax: Set Value: AT#FTPPORT=<value>

Get Value: AT#FTPPORT? or AT#VFTP, AT#VALL

#FTPMODE Enter FTP Mode

Definition: Define the FTP behavior for file transfer.

Values: **0** FTP Active Mode. **Default.**

1 FTP Passive Mode

Syntax: Set Value: AT#FTPMODE=<value>

Get Value: AT#FTPMODE? or AT#VFTP, AT#VALL

#FTPTYPE Specify Data Type

Definition: Before transferring files from a specified FTP server, the TCP/IP stack software must specify the type of data to be transferred within the FTP session.

Values: **A** FTP ASCII sessions

I FTP Binary sessions (upper case i letter). **Default.**

Note: When this value is set to A, all the data sent by the TCP/IP stack to the FTP server is made of 7 bits characters (NVT-ASCII: the MSB is set to 0). As a consequence binary data containing 8 bits characters will be corrupted during the transfer if the FTPTYPE is set to A.

Syntax: Set Value: AT#FTPTYPE=<value>

Get Value: AT#FTPTYPE? or AT#VFTP, AT#VALL

#FTPSERV Specify FTP Server Address

Definition: FTP server address. To connect to an FTP server to download files, the TCP/IP stack software must know the address of the FTP server that is to be used.

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or

Alphanumeric ASCII text string up to 120 characters if DNS is available. **No default**.

Syntax: Set Value: AT#FTPSERV=<value>

Get Value: AT#FTPSERV? or AT#VFTP, AT#VALL

#FTPUN Set FTP User Name

Definition: Before transferring files from a specified FTP server, the TCP/IP stack software must open an FTP session using a valid FTP user name.

Values: Alphanumeric ASCII text string up to 64 characters. **No default**.

Syntax: Set Value: AT#FTPUN=<value>

Get Value: AT#FTPUN? or AT#VFTP, AT#VALL

#FTPPW Set FTP Password

Definition: Before transferring files from a specified FTP server, the TCP/IP stack software must open an FTP session using a valid FTP password.

Values: Alphanumeric ASCII text string up to 64 characters. **No default**.

Syntax: Set Value: AT#FTPPW=<value>

Get Value: AT#FTPPW? or AT#VFTP, AT#VALL

#FTPGETFILENAME Set FTP Download File Name

Definition: In order to download a file from the FTP server, the TCP/IP stack software must know the name of the relevant file.

Values: Alphanumeric ASCII text string up to 120 characters. **No default**.

Syntax: Set Value: AT#FTPGETFILENAME=<value>

Get Value: AT#FTPGETFILENAME? or AT#VFTP, AT#VALL

#FTPGETPATH Set FTP Path

Definition: In order for the TCP/IP stack software to get a file from the FTP server, the TCP/IP stack software must know the path of the relevant file. For example, it could be: /list

Values: Alpha-numeric ASCII text string up to 120 characters. **No default**.
Note: Depending on the FTP server, the value can be used for getting a file from the root directory of the FTP server.

Syntax: Set Value: AT#FTPGETPATH=<value>

Get Value: AT#FTPGETPATH? or AT#VFTP, AT#VALL

#FTPPUTFILENAME Set FTP Upload File Name

Definition: In order for the TCP/IP stack software to upload a file to the FTP server, the TCP/IP stack software must know the name of the relevant file.

Values: Alpha-numeric ASCII text string up to 120 characters. **No default**.

Syntax: Set Value: AT#FTPPUTFILENAME=<value>

Get Value: AT#FTPPUTFILENAME? or AT#VFTP, AT#VALL

#FTPPUTPATH

Definition: In order for the TCP/IP stack software to upload a file to the FTP server, the TCP/IP stack software must know the path of the relevant file. For example, it could be: /list

Values: Alpha-numeric ASCII text string up to 120 characters. **No default**.
Note: Depending on the FTP server, the value can be used for getting a file from the root directory of the FTP server.

Syntax: Set Value: AT#FTPPUTPATH=<value>

Get Value: AT#FTPPUTPATH? or AT#VFTP, AT#VALL

#FTPGET Get Data from Server

Definition: This command, sent by the attached host, directs the TCP/IP stack to connect to the specified FTP server and to retrieve the specified file from this server. Once the operation is completed, the TCP/IP stack closes the FTP connection.

Once an IP link is established, the attached host can retrieve a file from an FTP server at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This command is similar to a GET operation (with an automatic connect/disconnect) issued by a standard FTP client on a PC. The TCP/IP stack handles the global FTP process by itself.

Note: Each <ETX> character present in the payload data of the FTP flow will be coded by the TCP/IP stack on the serial port as <DLE><ETX>. Each <DLE> character will be coded as <DLE><DLE>. The attached host must then decode the FTP flow to remove these escape characters.

Values: **Read Parameters:**
FTPGETFILENAME
FTPGETPATH
FTPPORT
FTPSERV
FTPTYPE
FPTMODE
FTPPW
FTPUN

Syntax: **AT#FTPGET**

Command	Possible Responses
AT#FTPGET Note: Start data reception	Ok_Info_DataBegin The switch from command to data mode. DATA **Note:** Data transmitted from the FTP server to the TCP/IP stack is sent over the serial port. <ETX> **Note:** Once the file transfer has finished, the TCP/IP stack sends an ETX character over the serial port to notify the attached host of the end of file transfer: switches from data to command mode. OK **Note:** The FTP process was successfully completed.
AT#FTPGET	#CME ERROR: 38027 **Note:** The address of the FTP server has not been resolved by the secondary DNS server. The TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong FTP server address has been filled in.
AT#FTPGET	#CME ERROR: <value> **Note:** The connection to the FTP server failed. See section 'Response messages and error codes.' If this error occurs once the data transfer started, it is preceded by an ETX character.

#FTPPUT Put Data to Server

Definition: This command sent by the attached host directs the TCP/IP stack to connect to the specified FTP server and to upload the data received on the serial port to the specified file on this server. Once the operation is completed, the TCP/IP stack closes the FTP connection.

Once an IP link is established, the attached host can send a file to a FTP server at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This command is similar to a PUT operation (with an automatic connect/disconnect) issued by a standard FTP client on a PC. The TCP/IP stack handles the global FTP put process by itself.

Note: The TCP/IP stack will interpret only an <ETX> character as the end of the file to be transferred if it's not preceded by a <DLE> character. As a consequence, the attached host must send <ETX> characters preceded by <DLE> characters, and it must also code <DLE> characters as <DLE><DLE>.

Values: **Read Parameters:**

FTPPUTFILENAME
FTPPUTPATH
FTPPORT
FTPSERV
FTPTYPE
FTPMODE
FTPPW
FTPUN

Syntax: **AT#FTPPUT**

Command	Possible Responses
AT#FTPPUT Note: Start sending data	OK_WaitingForData Note: TCP/IP stack is ready to send data from the serial port to the remote FTP server. TCP/IP stack then immediately transfers all the data sent by the attached host to the remote FTP server. To notify the TCP/IP stack that all data has been sent, the attached host must send the <ETX> character. <ETX> Notification from host for end of data: switches from data to command mode. OK Note: The FTP process was successfully completed.
AT#FTPPUT	#CME ERROR: 38027 Note: The address of the FTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong FTP server address has been filled in.
AT#FTPPUT	#CME ERROR: <value> Note: The connection to the FTP server failed. See section 'Response messages and error codes'. If an error occurs once the data transfer started, it is preceded by ***ETX*** .

#VFTP Display FTP Parameters

Definition: This command directs the TCP/IP stack to display all AT# parameters for the FTP client configuration.

Values: **Read Parameters:**

FTPSERV
FTPPORT
FTPUN
FTPPW
FTPGETFILENAME
FTPGETPATH
FTPPUTFILENAME
FTPPUTPATH
FTPTYPE
FTPMODE

Syntax: **AT#VFTP**

Command	Possible Responses
AT#VFTP Note: View FTP parameters	#FTPGETFILENAME: "" #FTPGETPATH: "." #FTPMODE: 0 #FTPPORT: 21 #FTPPUTFILENAME: "Testseb3.txt" #FTPPUTPATH: "." #FTPPW: "mypass" #FTPSERV: "mytestwebsite.com" #FTPTYPE: = 1 #FTPUN: "myname" OK

Chapter 5 - TCP Socket Services

Parameter Definitions

#DLEMODE Set DLE Mode

Definition: When using socket TCP, the attached host has the choice whether or not to code the ETX character.

Values:

0 When DLEMODE is set to 0, no specific process is needed on [ETX] characters.

It means that it is not possible for a host to request an end of connection or to receive a clear indication of the end of a connection from the TCP/IP stack.

1 When DLEMODE is set to 1, the [ETX] character means a request or an indication of end of connection.

As a consequence, [ETX] characters that belong to the payload data must be sent by the host on the serial port preceded by a DLE character. Similarly, ETX characters received by the TCP/IP stack from the Internet are sent to the host through the serial port preceded by a DLE character. **Default**.

id: {1,2}

Syntax: Set Value: AT#DLEMODE=id,<value>

Get Value: AT#DLEMODE=id or AT#VTCP=id, AT#VALL

#TCPPORT Set TCP Port for Remote Peer

Definition: To exchange data over TCP, the TCP/IP stack software must know the port of the remote peer used for the TCP session.

Values: From 1 to 5 digits (each digit between 0 and 9 inclusive). Note that numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header. **Default 0**.

id: {1,2}

Syntax: Set Value: AT#TCPPORT=id,<value>

Get Value: AT#TCPPORT=id or AT#VTCP, AT#VALL

#TCPSERV Set Address of Remote TCP Server

Definition: To exchange data over TCP, the TCP/IP stack software must know the address of the remote TCP server (or host) that is to be used.

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or

Alphanumeric ASCII text string up to 120 characters if DNS is integrated. **No default.**

id: {1,2}

Syntax: Set Value: AT#TCPSERV=id,value>

Get Value: AT#TCPSERV=id or AT#VTCP, AT#VALL

#LTCPTXDELAY Time Delay

Definition: This parameter determines the time delay introduced before sending a TCP frame that has not been entirely filled with user data. The time is entered in milliseconds, and it should be noted that a value of 0 initiates the sending of a TCP frame as soon as possible after the reception of a single character value from the host.

Values: Integer multiple of 20 between 0 and 32760 inclusive. **Default 100.**

id: {1,2}

Syntax: Set Value: AT#TCPTXDELAY=id,value>

Get Value: AT#TCPTXDELAY=id or AT#VTCP=id, AT#VALL

#LTCPSTART Open Listening Mode

Definition: This command, sent by the attached host, directs the TCP/IP stack to open a listening TCP connection on the specified TCP port.

Once an IP link is established, the attached host can open a listening TCP socket at any time (except when the TCP/IP stack software is already in a process using TCP resources).

The TCP connection will be active upon reception of a TCP connection request sent by a remote allowed TCP peer (TCPSERV) on the appropriate TCP port (TCPPORT).

Once opened, this TCP connection may be closed by the remote TCP peer or by the attached host by sending an ETX character on the serial port (depending on the DLEMODE parameter).

Notes:

The LTCP command can be aborted before an incoming TCP request has been received by issuing an <ETX> character on the serial port.

If the DLEMODE parameter is set to 1, the TCP/IP stack will interpret an <ETX> character only as a close request if a <DLE> character does not precede it. As a consequence, the attached host must send <ETX> characters preceded by <DLE> characters, and it must also code <DLE> characters in <DLE><DLE>. Similarly, each <ETX> character present in the payload data of the TCP frame will be coded by the TCP/IP stack on the serial port as <DLE><ETX>. Each <DLE> character will be coded as <DLE><DLE>. The attached host must then decode the TCP socket flow to remove these escape characters.

If the DLEMODE parameter is set to 0, the TCP/IP stack will never close the TCP connection (unless an error occurs).

Values: **Read Parameters:**
DLEMODE
TCPPORT
TCPSERV
TCPTXDELAY

id: {1,2}

Syntax: **AT#LTCPSTART=id, with id one of {1,2}**

Command	Possible Responses
AT#LTCPSTART=1 Note: Active listening mode	OK_Info_WaitingForData Note: This message signals that an allowed remote TCP peer has opened the TCP socket. The TCP connection is now opened. All the data from the attached host / remote TCP server is then immediately transferred by the TCP/IP stack to the remote TCP server / attached host. Depending on the DLEMODE value, the attached host may close this TCP connection by sending an ETX character. If the remote TCP server closes the connection, the TCP/IP stack issues an ETX character on the serial port.
AT#LTCPSTART=2	Ok_Info_SocketClosed OK Note: The TCP socket is closed.
AT#LTCPSTART=1	#CME ERROR: <value> Notes: An error has occurred during the TCP connection. This connection is being closed. If this error occurs once the TCP connection opened, it is preceded by an ETX character. See section 'Response messages and error codes'.

#LTCPSTOP Close Listening Mode

Definition: This command directs the TCP/IP stack to close a TCP listening mode (previously launched by the AT#LTCPSTART command.

Values: **Read Parameters:**
DLEMODE
TCPPORT
TCPSERV
TCPTXDELAY

id: {1,2}

Syntax: **AT#LTCPSTOP=id, with id one of {1,2}**

Command	Possible Responses
AT#LTCPSTOP=2 Note: Stop TCP listening	OK **Note:** The local listening port is closed in the TCP/IP stack.

#OTCP Open TCP Connection

Definition: This command sent by the attached host directs the TCP/IP stack to open a TCP connection to the specified TCP server. Once an IP link is established, the attached host can open a TCP connection at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This TCP connection may be closed by the remote TCP server or by the attached host via sending an ETX character on the serial port (**depending on the DLEMODE parameter**).

DLEMODE Value Notes:
Depending on the DLEMODE value, the attached host may close this TCP connection by sending an ETX character.

- If the DLEMODE parameter is set to 1, the TCP/IP stack will only interpret an <ETX> character as a close request if it's not preceded by a <DLE> character. As a consequence, the attached host must send <ETX> characters preceded by <DLE> characters, and it must also code <DLE> characters in <DLE><DLE>. Similarly, each <ETX> character present in the payload data of the TCP frame will be coded by the TCP/IP stack on the serial port as <DLE><ETX>. Each <DLE> character will be coded as <DLE><DLE>. The attached host must then decode the TCP socket flow to remove these escape characters.
- If DLEMODE is set to 0, the host cannot close the TCP connection (unless an error occurs).

If the remote TCP server closes the connection, the TCP/IP stack sends an ETX character on the serial port.

Values: **Read Parameters:**
DLEMODE
TCPPORT
TCPSERV
TCPTXDELAY

id: {1,2}

Syntax: **AT#OTCP=id, with id one of {1,2}**

Command	Possible Responses
AT#OTCP=1 Request TCP socket open	OK_Info_WaitingForData Note: This message signals that the TCP socket has been opened. All the data from the attached host / remote TCP server is then immediately transferred by the TCP/IP stack to the remote TCP server / attached host. Depending on the DLEMODE value, the attached host may close this TCP connection by sending an ETX character. If the remote TCP server closes the connection, the TCP/IP stack issues an ETX character on the serial port.
	OK_Info_SocketClosed OK Note: The TCP socket is closed.
AT#OTCP=2	#CME ERROR: 38027 Note: Address of FTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong FTP server address has been filled in.
AT#OTCP=2	#CME ERROR: <value> Note: An error has occurred during the TCP connection. This connection is being closed. If this error occurs once the TCP connection opened, it is preceded by an ETX character. See section 'Response messages and error codes'.

#VTCP Display TCP Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the TCP socket configuration.

Values: **Read Parameters:**
DLEMODE
TCPPORT
TCPSERV
TCPTXDELAY

id: {1,2}

Syntax: **AT#VTCP=id, with id one of {1,2}**

Command	Possible Responses
AT#VTCP=2 Note: View TCP parameters of the TCP socket 2	#DLEMODE: 2,1 #TCPSERV: 2,"" #TCPPORT: 2,0 TCPTXDELAY: 2,100 OK

Chapter 6 – UDP Socket Services

Parameters Definitions

#UDPPORT UDP Port Number

Definition: If the UDP session is initiated in listen mode, use the local UDP port number.

If the UDP session is initiated in active mode, use the remote UDP port number.

Values: From 1 to 5 numeric digits (0 to 9). **Default is 0.**

Notes:

- Numbers above 65,535 are illegal as the port identification fields are 16-bits long in the IDP header.
- Port number 0 is illegal.

Syntax: Set Value: AT#UDPPORT=<Value>

Get Value: AT#UDPPORT? or AT#VUDP, AT#VALL

#UDPSERV UDP Server

Definition: If the UDP session is initiated in listen mode, use the IP address filter. This means that the remote must have a defined UDPSERV IP address.

If the UDP session is initiated in active mode, use the remote IP address.

Note: No IP filter is applied if the parameter value is **255.255.255.255**

Values: 32-bit number is dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or

Alphanumeric ASCII text string up to 120 characters if DNS is integrated.

There is **no default** value for this parameter.

Syntax: Set Value: AT#UDPSERV=<Value>

Get Value: AT#UDPSERV? or AT#VUDP, AT#VALL

#UDPTXDELAY UDP Data Delay

Definition: This parameter determines the delay before sending an UDP datagram that has not been entirely filled with user data. The delay is expressed in milliseconds.

The 0 value initiates the sending an UDP datagram as soon as possible after the reception of a single character value from the host.

Values: Integer, multiple of 20 and between 0 and 32760 inclusive.

Default value is 100.

Syntax: Set Value: AT#UDPTXDELAY=<Value>

Get Value: AT#UDPTXDELAY? or AT#VUDP, AT#VALL

#LUDPSTART Open an UDP Session in Listen Mode

Definition: Once an IP link is established, and if no other TCP/IP stack resource is active, this command sent by the attached host directs the TCP/IP stack to initiate the UDP session in listen mode on the specified UDP local port UDPPORT. Depending on the UDPSERV parameter content, there are two possible combinations:

- If UDPSERV is 255.255.255.255, the IP address filter feature is turned off. The UDP session will be effective upon reception of the first datagram, supplying the remote port number and the remote IP address of the session. The host can then transmit to the (remote port or remote IP) address.
- If UDPSERV holds a valid IP address (or existing/found alphanumeric IP address), only the remote with the UDPSERV IP address will be authorized to send datagrams to this UDP socket (i.e., the IP address filter feature is activated). The UDP session will be effective upon reception of the first datagram, supplying the remote port number of the session. The host can then transmit to the (remote port, UDPSERV) address. Datagrams can then only be received from this (remote port, UDPSERV) address.

As long as the first datagram is not received, UDPSTOP can be issued to cancel the UDP session. Once the first datagram has been received, the only way to stop the UDP session for the host is to issue an <ETX> character.

Note: The TCP/IP stack will only interpret an <ETX> character as a close request if it is not preceded by a <DLE> character. As a consequence, an <ETX> character must be sent (even in payload data) as <DLE><ETX>, and <DLE> character as <DLE><DLE>. The attached host must then decode the UDP socket flow to remove these escape characters.

Values: **Read parameters:**
UDPPORT
UDPSERV
UDPTXDELAY

Syntax: **AT#LUDPSTART**

Command	Possible Responses
AT#LUDPSTART Note: Active listen mode	OK_Info_WaitingForData Notes: This message signals that the remote allowed UDP peer has sent its first datagram to the local UDP socket. The UDP connection is now effective. All data from the attached host / remote UDP peer is immediately transferred by the TCP/IP stack to the remote UDP peer / attached host. Attached host may close this UDP session by sending an <ETX> character. If an error occurs, the TCP/IP stack issues and <ETX> character on the serial port before sending the appropriate error message.
AT#LUDPSTART	OK_Info_WaitingForData ... ... <ETX> OK_Info_SocketClosed OK Note: The UDP socket is closed.
AT#LUDPSTART	#CME ERROR: <value> Note: An error has occurred during the UDP session creation. If this error occurs once the UDP session is effective, it is preceded by an <ETX> character.

#LUDPSTOP Close a Listening Mode UDP Session

Definition: This command directs the TCP/IP stack to close a UDP listening mode session (previously launched by the AT#LUDPSTART command).

Values: **Read parameters:**
UDPPORT
UDPSERV
UDPTXDELAY

Syntax: AT#LUDPSTOP

Command	Possible Responses
AT#LUDPSTOP Note: Stop UDP listening	OK **Note:** The local listening port is closed in the TCP/IP stack.

#OUDP Open an Active UDP Session

Definition: Once an IP link is established, and if no other TCP/IP stack resource is active, this command sent by the attached host directs the TCP/IP stack to initiate an UDP session in active mode on the specified UDP remote port UDPPORT to the specified remote IP address UDPSERV.

The host can then transmit to the UDPPORT or UDPSERV address. Datagrams can only be received from this UDPPORT or UDPSERV address.

The host can stop the UDP session by issuing an <ETX> character.

Note: The TCP/IP stack will only interpret an <ETX> character as a close request if it is not preceded by a <DLE> character. As a consequence, an <ETX> character must be sent (even in payload data) as <DLE><ETX>, and <DLE> character as <DLE><DLE>. The attached host must then decode the UDP socket flow to remove these escape characters.

Values: **Read parameters:**
UDPPORT
UDPSERV
UDPTXDELAY

Syntax: **AT#OUDP**

Command	Possible Responses
AT#OUDP Note: Request opening of UDP socket	OK_Info_WaitingForData Notes: This message signals that the UDP socket has been opened. All data from the attached host / remote UDP peer is immediately transferred by the TCP/IP stack to the remote UDP peer / attached host. The attached host may close this UDP session by sending and <ETX> character. If an error occurs, the TCP/IP stack issues and <ETX> character on the serial port before sending the appropriate error message.
AT#OUDP	OK_Info_WaitingForData ... ... <ETX> OK_Info_SocketClosed OK Note: The UDP socket is closed.
AT#OUDP	#CME ERROR: 38027 Note: The address of the remote UDP peer has not been resolved by the secondary DNS server. The TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong remote UDP peer address has been filled in.
AT#OUDP	#CME ERROR: <value> Note: An error has occurred during the UDP session creation. If this error occurs once the UDP session is effective, it is preceded by an <ETX> character.

#VUDP Display UDP Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the UDP socket configuration.

Values: **Read parameters:**
UDPPORT
UDPSERV
UDPTXDELAY

Syntax: **AT#VUDP**

Command	Possible Responses
AT#VUDP Note: View UDP parameters	#UDPSERV: “” #UDPPORT: 0 #UDPTXDELAY: 100 OK

Chapter 7 – PING Services

Parameters Definitions

#PINGDELAY PING Delay Time

Definition: This command sets the waiting delay, in seconds, before an echo request is considered as a ***no reply***. It is also the delay between two echo requests (if PINGNUM > 1).

Values: From 1 to 255 inclusive. **Default value is 1**.

Syntax: Set Value: AT#PINGDELAY=<Value>

Get Value: AT#PINGDELAY? or AT#VPING, AT#VALL

#PINGNUM Number of PING Requests

Definition: This command sets the number of PING echo requests to issue to PINGREMOTE.

Values: From 1 to 255 inclusive. **Default value is 4.**

Syntax: Set Value: AT#PINGNUM=<Value>

Get Value: AT#PINGNUM? or AT#VPING, AT#VALL

#PINGREMOTE IP Address of PING Request

Definition: Sets the IP address or alphanumeric ASCII text string, up to 120 characters long if DNS is available.

Values: 32-bit number is dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or

Alphanumeric ASCII text string up to 120 characters if DNS is integrated. **No default.**

Syntax: Set Value: AT#PINGREMOTE=<Value>

Get Value: AT#PINGREMOTE? or AT#VPING, AT#VALL

#PING Start PING Request

Definition: Once an IP link is established, and if no other TCP/IP stack resource is active, this command sent by the attached host directs the TCP/IP stack to start PING requests.

Values: **Read parameters:**
PINGDELAY
PINGNUM
PINGREMOTE

Syntax: **AT#PING**

Command	Possible Responses
AT#PING	"wavecom.com" is alive : time = 900 ms "wavecom.com" is alive : time = 900 ms "wavecom.com" is alive : time = 800 ms "wavecom.com" is alive : time = 800 ms "wavecom.com" is alive : time = 1000 ms NO answer from "wavecom.com" "wavecom.com" is alive : time = 900 ms "wavecom.com" is alive : time = 800 ms "wavecom.com" is alive : time = 900 ms "wavecom.com" is alive : time = 900 ms OK Note: TCP/IP stack sends PINGNUM = 10 requests
AT#PING	#CME ERROR: 38027 Note: The address of the remote has not been resolved by the DNS servers. The TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong remote address has been filled in.
AT#PING	#CME ERROR: <value> Note: An error has occurred during the PING requests.

#VPING Display PING Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the PING configuration.

Values: **Read parameters:**
PINGDELAY
PINGNUM
PINGREMOTE

Syntax: **AT#VPING**

Command	Possible Responses
AT#VPING Note: View PING parameters	#PINGDELAY: 1 #PINGNUM: 4 #PINGREMOTE: " " OK

Chapter 8 – Miscellaneous Commands

#VVERSION Display Software Version

Definition: This command directs the TCP/IP stack to display the software version.

Values: No TCP/IP parameters.

Syntax: **AT#VVERSION**

Response Syntax: **<Filename> - <File size> - <Date and time of generation>**

Command	Possible Responses
AT#VVERSION Note: Request TCP/IP stack version	#VERSION: “eDsoft-W302_V01.00 67826 Mar 18 2003 18:42:19” OK

#VSTATE Display Current Status

Definition: This command directs the TCP/IP stack to display the current status of the device.

Values: No TCP/IP parameters.

Syntax: **AT#VSTATE**

Command	Possible Responses
AT#VSTATE	#STATE: “IDLE” OK Note: Idle state.
AT#VSTATE	#STATE: “DIALING” OK Note: Dialing the ISP. Not yet connected.
AT#VSTATE	#STATE: “AUTHENTICATING” OK Note: Connection. Not yet PPP negotiated. (PPP OK message).
AT#VSTATE	#STATE: “CONNECTED” OK Note: Connected to Internet. An IP address has been attributed to the TCP/IP stack.
AT#VSTATE	#NO SERVICE Note: No radio network connection made.
AT#VSTATE	#STATE: “DISCONNECTING” Note: In the process of disconnecting from the current communication process.
AT#VSTATE	#STATE: “CHECKING” Note: Either, in the process of setting up an outgoing communication session over GSM or GPRS, or in the process of answering an incoming call (manually or automatically) over GSM.

#VALL Display All Parameters

Definition: This command directs the TCP/IP stack to display all AT# parameters. The parameters are displayed by blocks of categories separated by a <CR><LF> sequence, all at the same time.

Values: Read parameters. All parameters.

Syntax: **AT#VALL**

Command	Possible Responses
AT#VALL	eDsoft-W302-V03.00 #ANSWERMODE: 0 #CALLBACKTIMER: 2 #CALLSCREENNUM: "0" #DIALN1: " " #DIALN2: " " #DIALSELECT: 1 #GPRSMODE: 1 #PHYTIMEOUT: 15 #REDIALCOUNT: 5

		#REDIALDELAY: 5 #RINGCOUNT: 0 #ISPUN: " " #ISPPW: " " #PPPMODE: 1 #PPPMYIP: "0.0.0.0" #PPPPEERIP: "0.0.0.0" #PPPSERVUN: " " #PPPSERVPW: " " #GPRSCID: 1 #APNPW: "access" #APNSERV" "a2bouygtel.com" #APNUN: "a2b" #DNSSERV1: "0.0.0.0" #DNSSERV2: "0.0.0.0" #FTPSERV: “mytestwebsite.com” #FTPPORT: 21 #FTPMODE: 0 #FTPUN: “mylogin” #FTPPW: “mypassword” #FTPTYPE: = 1 #FTPGETFILENAME: “” #FTPGETPATH: “.” #FTPPUTFILENAME: “Testseb3.txt” #FTPPUTPATH: “.” #POP3HEADERMODE: 1 #POP3PORT: 110 #POP3PW: “ ” #POP3SERV: “ ” #POP3UN: “ ” #DOMAIN: “a2bouygtel.com” #SENDERADDR: “toto@bouygtel.com” #SENDERNAME: “toto” #SMTPPORT: 25 #SMTPPW: “ ” #SMTPSERV: “smtp.a2bouygtel.com” #SMTPUN: “ ” #BODY1: “ ” #CCREC1: “ ” #REC1: “lesavecom.com” #SUBJ1: “Test” #BODY2: “ ” #CCREC2: “ ” #REC2: “ ” #SUBJ2: “ ” #BODY3: “9.1234567890abcd” #CCREC3: “” #REC3: “ ” #SUBJ3: “ ” #DELMODE: 1,1 #TCPSERV: 1,“” #TCPPORT: 1,0 #TCPTXDELAY: 1,100 #DELMODE: 2,1 #TCPSERV: 2,“” #TCPPORT: 2,0 #TCPTXDELAY: 2,100 #UDPPORT: 0 #UDPSERV: " " #UDPTXDELAY: 100 #PINGDELAY: 1 #PINGNUM: 4 #PINGREMOTE: " " OK

#DELFLASH Erase Flash Memory Parameter Values

Definition: This command erases the contents of parameter flash memory. Used prior to a "hard reset" of the modem, it results in the default values being written into the parameter memory.

Note: The current parameter values remain visible until the "hard reset" has completed.

Values: No parameter values.

Syntax: **AT#DELFLASH**

Command	Possible Responses
AT#DELFLASH Note: Delete flash memory contents	OK

Chapter 9 - Response Messages and Error Codes

Response Messages

Standard AT Messages		
Numeric	**Verbose**	**Description**
0	OK	Operation or command success
3	NO CARRIER	No physical layer connection
7	BUSY	Destination busy
8	NO ANSWER	No answer from destination
4	ERROR	Operation or command unsuccessful
2	RING	Incoming call indication
10	CONNECT 300	Physical layer connected at 300 baud
11	CONNECT 1200	Physical layer connected at 1200 baud
12	CONNECT 1200/75	Physical layer connected at 1200/75 baud
13	CONNECT 2400	Physical layer connected at 2400 baud
14	CONNECT 4800	Physical layer connected at 4800 baud
15	CONNECT 9600	Physical layer connected at 9600 baud
16	CONNECT 14400	Physical layer connected at 14400 baud

Information Messages		
Numeric	**Verbose**	**Description**
1025	OK_Info_DataBegin	Start of data
1028	OK_Info_WaitingForData	Send data
3074	OK_Info_SocketClosed	Socket connection closed successfully
3072	OK_Info_NoMail	No mail to retrieve on server
3073	OK_Info_Mail	Mail ready to be retrieved on server
3077	OK_Info_PPP	PPP connection successful
3086	OK_Info_GPRSActivation	GPRS connection successful

Error Codes

Error Codes	
Numeric	**Description**
34817	Bad command : Unknown command
34819	Bad command : Syntax error
34824	Bad command : EEPROM write failed
34881	Bad command : Command to long
34882	Bad command : Bad command argument value
35840	Physical layer : Modem is already running
35841	Physical layer : GPRS session lost
35862	Physical layer : Timeout, no activity on network connection
35865	Physical layer : Module is not attached to the network
35866	Physical layer : Invalid event during activation process
35867	Physical layer : Physical layer connection is currently not active
35868	Physical layer : GPRS connection aborted
35869	Physical layer : Invalid incoming call type
35870	Physical layer : Incoming call CLI not provided
36872	IP Connectivity library internal error : internal resource unavailable.
36929	IP Connectivity library : Bad parameter configuration attempt
37120	IP Connectivity library : PPP negotiation failed (client configuration)
37121	IP Connectivity library : PPP negotiation failed (server configuration)
37122	IP Connectivity library : Another internal application is already running
37123	IP Connectivity library : Service is running. Unable to set parameter
37952	Distant : TCP session closed (TCP Context cancelled)

Error Codes	
Numeric	**Description**
37964	Distant : No response from server
37966	Distant : TCP session closed by peer (FIN received from peer)
38016	Distant : Open session attempt failed
38017	Distant : Data send attempt failed
38018	Distant : Close session attempt failed
38022	Distant : Change Directory attempt failed
38023	Distant : File deletion attempt failed
38024	Distant : Data retrieve attempt failed
38025	Distant : Email retrieve attempt failed
38026	Distant : Email header receive failed
38027	Distant : No answer from DNS servers or the domain name resolution could not be completed by the server.
38028	Distant : Sender email address rejected by server
38029	Distant : Recipient email address rejected by server
38030	Distant : CC Recipient email address rejected by server
38031	Distant : Email body send request rejected by server
38080	Distant : Username rejected by server
38081	Distant : Password rejected by server
38980	SmartStack : PPP timeout (client configuration)
38981	SmartStack : PPP timeout (server configuration)
49153	Internal error : Open data flow request failed
49154	Internal error : Close data flow request failed
49155	Internal error : Open GPRS session request failed
49156	Internal error : GPRS authentication failed
49157	Internal error : GPRS get IPCP information request failed
49158	Internal error : Open flow confirmation not received

Appendix A – Getting Started Examples

AT+WOPEN Start TCP/IP Stack

Command	Responses
AT+WOPEN=1 *Note: Start the TCP/IP stack*	OK

AT+WOPEN=0
The TCP/IP stack is not opened. In some instances, this may be the default setting.

AT#VVERSION Get Software Version

AT#VVERSION *Note: Check the TCP/IP stack software version*	#VERSION: "eDsoft-W302_V01.00 67826 Mar 18 2003 18:42:19"

AT#PPPMODE Set Parameter Value

Command	Responses
AT#PPPMODE=1 Note: Set a parameter value	OK
AT#PPPMODE? Note: Request a parameter value	#PPPMODE: 1 OK

AT+CPIN GSM Network Registration

Command	Responses
AT+CPIN=xxxx Note: Enter the pin code	OK
AT+CREG=1 Note: Ask for registration state	OK +CREG: 1

AT+CGREG GSPRS Network Registration State

Command	Responses
AT+CGREG=1 Note: Ask for GPRS registration state	OK +CGREG: 0
AT+CGATT=1 Note: GPRS attach	+CGREG: 2 +CGREG: 1

Connect to the Internet

Connection to the Internet Service Provider Using GSM Data: AT#CONNECTIONSTART

Commands	Responses
AT#GPRSMODE=0 *Note: Activate the GSM data mode*	OK
AT#DIALN1=”0860000000” *Note: Set the ISP phone number*	OK
AT#ISPUN=”login” *Note: Set the ISP login*	OK
AT#ISPPW=”password” *Note: Set the ISP password*	OK
AT#DNSSERV1=”xxx.xxx.xxx” *Note: Set the DNS server address (coming with the ISP subscription)*	OK
AT#CONNECTIONSTART *Note: Launch the ISP connection*	DIALING 0860000000 *Note: The remote modem answers a call* *CONNECT 9600* *213.30.30.30* *Note: Dynamic IP address attributed by the network* *OK_Info_PPP* *Note: The PPP link is established*
AT#CONNECTIONSTOP *Note: When required, stop the Internet connection*	OK

Connection to the Internet Service Provider Using GPRS: AT#CONNECTIONSTART

Commands	Responses
AT#GPRSMODE=1 Note: Activate the GPRS mode	OK
AT#APNSERV=”apnserver.com” Note: Set the APN server (GPRS provider)	OK
AT#APNUN=”login” Note: Set the APN login	OK
AT#APNPW=”password” Note: Set the APN password	OK
AT#CONNECTIONSTART Note: Launch the GPRS connection	213.30.30.30 Note: Dynamic IP address attributed by the network OK_Info_GprsActivation Note: The GPRS/IP link is established
AT#CONNECTIONSTOP Note: When required, stop the Internet connection	OK

Sending/Retrieving Email

Sending an Email: AT#PUTMAIL

Commands	Responses
AT#SMTPSERV= ”smtp.domain.com” *Note: SMTP server used*	OK
AT#DOMAIN= ”domain.com” *Note: Domain name*	OK
AT#SENDERNAME= ”Test module” *Note: Sender name*	OK
AT#SENDERADDR= ”module@domain.com” *Note: Sender email address*	OK
AT#REC1= recipient@domain.com *Note: Recipient email address*	OK
AT#CCREC1= ccrecipient@domain.com *Note: Carbon Copy recipient*	OK
AT#SUBJ1= “Email Subject” *Note: Email Subject*	OK
AT#PUTMAIL *Note: Send an email (type the email text and then the end sequence)*	OK_Info_WaitingForData *Note: The software is ready to receive incoming data (not echoed)* *At the end of date, the* [CR][LF] . [CR][LF] sequence ends the email. This sequence can be sent by a keyboard using: ENTER CTRL+ENTER . ENTER CTRL+ ENTER
Note: End sequence sent	OK *Note: The email is successfully sent*

Retrieving an Email: AT#GETMAIL

Commands	Responses
AT#POP3SERV= ”pop3.domain.com” Note: POP3 server used	OK
AT#POP3UN= ”module@domain.com” Note: POP3 username (not always the complete email address). It is the POP3 login	OK
AT#POP3PW = ”password” Note: POP3 password	OK
AT#GETMAIL Note: Retrieve an email	OK_Info_Mail Note: The software switches from command mode to data mode for receiving the email content. Data Data . Note: At the end of data, the [CR][LF] . [CR][LF] sequence notifies the end of the data mode OK Note: The email is successfully retrieved

FTP: Download / Upload Files

Upload a file to an FTP Server: AT#FTPPUT

Commands	Responses
AT#FTPSERV=”pop3.domain.com” Note: FTP server used	OK
AT#FTPUN= ”ftplogin” Note: FTP username	OK
AT#FTPPW = ”ftppassword” Note: FTP password	OK
AT#FTPPUTFILENAME=”upload.text” Note: Name of the file that will be written in the FTP server	OK
AT#FTPPUTPATH=”.” Note: Path in the server where the file will be written	OK
AT#FTPPUT Note: FTP put	OK_Info_WaitingForDate Note: Switch from command to data mode. The host can send the data that will compose the file. (Data not echoed). To notify the end of data, the host has to send the [ETX] character (CTRL+C on the keyboard). This character is echoed. OK
Note: End of data notified	OK

Download a file from an FTP Server: AT#FTPGET

Commands	Responses
AT#FTPSERV=”pop3.domain.com” Note: FTP server used	OK
AT#FTPUN= ”ftplogin” Note: FTP username	OK
AT#FTPPW = ”ftppassword” Note: FTP password	OK
AT#FTPGETFILENAME=”upload.text” Note: Name of the file stored in the FTP server	OK
AT#FTPGET Note: FTP get	OK_Info_DataBegin Note: Switch from command to data mode. The data is sent over the serial port Note: To end of data is notified by the [ETX] character sent over the serial port. It switches from data to command mode. Character is echoed. OK

TCP Socket

Open a TCP socket between machines. One machine acts as a caller (TCP client); the other acts as a listener (TCP server). Both machines have to be connected to the Internet and set to the same TCP port.

Act as a TCP Server: AT#LTCPSTART

Commands	Responses
AT#TCPSERV=1,"255.255.255.255" Note: No filter of the incoming TCP client	OK
AT#TCPPORT=1,"23" Note: TCP port between the TCP client and the TCP server must be the same	OK
AT#LTCPSTART=1 Note: Launch the listening mode; waiting for an incoming TCP connection from a TCP client	OK OK_Info_WaitingForDate Note: Message sent over the serial port in case of successful TCP socket opening (Telnet for example in the IP address)
Data Note: Data flow is bidirectional	Data Note: Data flow is bidirectional
Note: The socket can be closed locally by the attached host sending an [ETX] character (CTRL+C on the keyboard)	OK
	Note: The socket can be closed by the remote OK_Info_SocketClosed OK

Act as a TCP Client: AT#OTCP

Commands	Responses
AT#TCPSERV=1,"xxx.xxx.xxx.xxx" Note: No filter of the incoming TCP client	OK
AT#TCPPORT=1,"23" Note: TCP port between the TCP client and the TCP server must be the same	OK
AT#OTCP=1 Note: Open as a TCP client, a socket TCP with remote TCP server	OK_Info_WaitingForDate Note: Message notifying the socket opening and the switch in data mode
Data Note: Data flow is bidirectional	Data Note: Data flow is bidirectional
Note: The socket can be closed locally by the attached host sending an [ETX] character (CTRL+C on the keyboard)	OK
	Note: The socket can be closed by the remote OK_Info_SocketClosed OK

Note: The closing of the socket can be performed either locally or remotely.

Appendix B – Detailed Examples

Serial Communication

This example shows the serial port configuration and syntax for communication between the modem and the attached equipment.

Configuration

Serial port data rate	Apply to the software the same serial baud rate as the one running on the attached equipment. AT+IPR parameter
Flow Control	Apply to the software the same flow control mechanism as the one running on the attached equipment: either hardware (RTS/CTS) or none. AT+IFC parameter
Serial echo	Enable or disable the echo over the serial port of the characters received by the software. ATE parameter
Result code suppression	This parameter selects the activation or suppression of the response codes returned by the software: response or no response. ATQ parameter
Response format	This parameter selects the format of the response codes returned by the software: numeric or alpha-numeric responses. ATV parameter
Note	Data format is: 8 data bits, no parity, 1 stop bit.

Output

Viewing a parameter	After typing `AT#`*`Mnemonic`*`?` The modem sends the following sequence over the serial port: if the ATQ configuration allows the response message sending, and whatever is the ATV value (0 or 1) #Mnemonic: value[CR][LF] [CR][LF]OK[CR][LF] the value is between “” if it is a string.
Indication/ Response message from the modem	The modem sends indications and return codes over the serial port with the following syntax: if the ATQ configuration allows the response message sending, **ATV1** **ATV0** [CR]numeric code[CR] [CR][LF] For an #CME ERROR, the format of the response is: Indication message in text [CR]#CME ERROR: Value [CR][LF] [CR][LF]
Writing a parameter	After typing *AT#Mnemonic=value*, the software sends the following sequence over the serial port (upon completion). Value shall be between “” if it is a string. The “” are not required if the value is an integer. if the ATQ configuration allows the response message sending, **ATV1** **ATV0** [CR][LF] [CR]0[CR] OK[CR][LF]
Note	A 1 second delay shall be introduced before each AT# command issued for writing parameters value.

Serial Communication Continued

Command Mode / Data Mode for Email

Email service: SMTP through AT#PUTMAIL ENTRY FLOW	**Enter in DATA mode:** The “Ok_Info_WaitingForData” message sent to the host on the serial port indicates the entry in DATA mode. Modem ready to receive data from host. This data will form the body of the email	**Exit DATA mode:** 1. The host ends the DATA mode by sending the [CR][LF].[CR][LF] sequence on the serial port 2. The modem ends the DATA mode if an error occurs by sending an error message to the host (this message is not preceded by the [CR][LF].[CR][LF] sequence)
Email service: POP3 through AT#GETMAIL OUTPUT FLOW	**Enter in DATA mode:** The “Ok_Info_Mail” message sent to the host on the serial port indicates the entry in DATA mode. The modem then immediately sends the data received from the POP3 server to the host.	**Exit DATA mode:** 1. The modem ends the DATA mode by sending the [CR][LF].[CR][LF] sequence on the serial port. This sequence is sent in case of a positive result or if an error occurs. 2. It's not possible for host to exit DATA mode

Command Mode / Data Mode for FTP

FTP service : FTP through AT#FTPPUT ENTRY FLOW	**Enter in DATA mode:** The “Ok_Info_WaitingForData” message sent to the host on the serial port indicates the entry in DATA mode. The modem is ready to receive the data from the host for sending it. This data will form the file on the FTP server.	**Exit DATA mode:** 1. The host ends the DATA mode by sending the [ETX] character (not preceded by [DLE]) on the serial port. 2. The modem ends the DATA mode if an error occurs by sending an error message to the host (this message is not preceded by the [ETX] character). [ETX] is CTRL+C in a keyboard. [DLE] is CTRL+P in a keyboard.
FTP service : FTP through AT#FTPGET OUTPUT FLOW	**Enter in DATA mode:** The “Ok_Info_DataBegin” message sent to the host on the serial port indicates the entry in DATA mode. The modem then immediately sends the data received from the FTP server to the host.	**Exit DATA mode:** 1. The modem ends the DATA mode by sending the [ETX] character (not preceded by [DLE]) on the serial port. This character is sent in case of a positive result or if an error occurs. 2. It's not possible for host to exit DATA mode. [ETX] is CTRL+C in a keyboard. [DLE] is CTRL+P in a keyboard.

Command Mode / Data Mode for TCP Services: [ETX] means END of TCP connection

When DLEMODE is set to 1, the [ETX] character means a request or an indication of end of connection.

As a consequence, [ETX] characters that belongs to the payload data must be sent by the host on the serial port preceded by a [DLE] character. Similarly [ETX] characters received by the modem from the Internet are sent to the host through the serial port preceded by a [DLE] character

Socket TCP service : TCP socket through AT#LTCPSTART or AT#OTCP ENTRY FLOW	**Enter in DATA mode:** The "Ok_Info_WaitingForData" message sent to the host on the serial port indicates the entry in DATA mode. The modem is ready to receive the data from the host. This data will be sent to the remote TCP peer.	**Exit DATA mode:** 1. The host ends the DATA mode by sending the [ETX] character (not preceded by [DLE]) on the serial port. 2. The modem ends the DATA mode if an error occurs or if the TCP peer disconnects by sending an error message to the host (this message is preceded by the [ETX] character) [DLE] is CTRL+P in a keyboard. [ETX] is CTRL+C in a keyboard.
Socket TCP service : TCP socket through AT#LTCPSTART or AT#OTCP OUTPUT FLOW	**Enter in DATA mode:** The "Ok_Info_WaitingForData" message sent to the host on the serial port indicates the entry in DATA mode. The modem then immediately sends the data received from the remote TCP peer to the host.	**Exit DATA mode:** 1. The host ends the DATA mode by sending the [ETX] character (not preceded by [DLE]) on the serial port. 2. The modem ends the DATA mode if an error occurs or if the TCP peer disconnects by sending a message to the host (this message is preceded by the [ETX] character) [ETX] is CTRL+C in a keyboard. [DLE] is CTRL+P in a keyboard.

Command Mode / Data Mode for TCP Services: ETX is a character

When DLEMODE is set to 0, no specific process is needed on [ETX] characters.

It means that it is not possible for a host to request an end of connection or to receive a clear indication of end of connection from the modem.

Socket TCP service : TCP socket through AT#LTCPSTART or AT#OTCP ENTRY FLOW	**Enter in DATA mode:** The “Ok_Info_WaitingForData” message sent to the host on the serial port indicates the entry in DATA mode. The modem is ready to receive the data from the host. This data will be sent to the remote TCP peer.	**Exit DATA mode:** 1. The host can't end the DATA mode. 2. The modem ends the DATA mode if an error occurs or if the TCP peer disconnects by sending a message to the host (this message is not preceded by the [ETX] character)
Socket TCP service : TCP socket through AT#LTCPSTART or AT#OTCP OUTPUT FLOW	**Enter in DATA mode:** The “Ok_Info_WaitingForData” message sent to the host on the serial port indicates the entry in DATA mode. The modem then immediately sends the data received from the remote TCP peer to the host.	**Exit DATA mode:** 1. The host can't end the DATA mode. 2. The modem ends the DATA mode if an error occurs or if the TCP peer disconnects by sending a message to the host (this message is not preceded by the [ETX] character)
Socket TCP service : TCP socket through AT#LTCPSTART or AT#OTCP	When DLEMODE is set to 1, the [ETX] character means a request or an indication of end of connection. As a consequence, [ETX] characters that belongs to the payload data must be sent by the host on the serial port preceded by a [DLE] character. Similarly [ETX] characters received by The modem from the Internet are sent to the host through the serial port preceded by a [DLE] character. [ETX] is CTRL+C in a keyboard. [DLE] is CTRL+P in a keyboard.	

Configuring the TCP/IP Stack through the Serial Port

This example shows the communication format for configuring the TCP/IP stack parameters via the serial port.

Description: The modem contains a number of parameters that are used to control the transmission sessions over the Internet. All parameters are stored in the internal memory present on the hardware platform. Because the modem parameters are stored in nonvolatile memory, they retain their values until they are changed by a new command.

The AT# commands allow to configure the modem and the AT commands allow to configure the software.

Syntax: The syntax of all commands to be sent from the attached equipment to the software through the serial port is:

AT#<Mnemonic>=<Value>(CR)

AT# is a prefix required for all the commands. This prefix is not case sensitive and consists of three ASCII fields, whose values may differ depending on the case used.

<Mnemonic> is an ASCII text string indicating which parameter is to be changed. This string is not case-sensitive.

Values:

- An ASCII string representing the new value of the parameter to be changed. In this case, the value has to be entered between quotes; For example: AT#POP3SERV=”name.domain.com”
- A numeric value representing the new value of the parameter to be changed. In this case, the value can be directly entered without quotes after the =. For example: AT#FTPPORT=21

Note: (CR) is a carriage return (ASCII 13) character, signaling the end of the command line.

It is possible to set parameter values while either:

- **Offline**; i.e., not in communication with the attached network). The new parameter values are immediately stored in the internal memory.
- **Online**; i.e., in communication with the attached network, only if an application process (FTP, email, etc.) is not currently active. The new parameter values are directly stored into the internal memory.

  While offline, or online and not during a process, all characters are echoed over the serial port according to the echo configuration (ATE parameter)

Requesting the Parameter values

Listing the value of all parameters	The '**AT#VALL**' command sent over the serial port directs the modem to return the values of ALL parameters sequentially over the serial port.
AT# Command	AT#VALL
Listing parameter values sorted by category	Some additional commands allow the user to list a group of parameter values by category. The flow control mechanism must match the one running on the attached equipment: hardware or none.
AT# Commands	AT#VPHY – AT#VPPP – AT#VPOP3 – AT#VSMTP – AT#VMAIL1/AT#VMAIL2/AT#VMAIL3 – AT#VDNS – AT#VFTP – AT#VTCP
Listing the value of one parameter AT# Command	The user can read the value of individual parameters. AT#parameter? Example: AT#POP3SERV? Will display the POP3SERV value : #POP3SERV = Value Value between “” if it is a string OK The software is ready

Notes:
The response format will depend on the setting of the ATV parameter setting.
The displaying of the response codes will depend on the setting of the ATQ parameter setting.

Configuring Flow Control over the Serial Port

The following description explains how to configure flow control for communication between the modem and the attached equipment.

Description: The flow control is managed by the software. It is configured through the AT command: AT+IFC

Using the DNS Service

This example shows how to use the DNS client embedded in the modem in order to use symbolic names instead of IP addresses.

Description: The DNS servers available on the Internet or within an Intranet are used to resolve a symbolic hostname (like mail.domain.com) into an IP address (192.168.255.250).
DNS provides a means of using user-friendly and easy to remember names for email, TCP, FTP servers rather than using the standard IP addressing format. Moreover these servers can switch to a different IP addressed destination without requiring any change on the symbolic name configuration.

Configuration:

Primary DNS	The IP address of the primary DNS server must be stored in dotted decimal notation (xxx.xxx.xxx.xxx). This address is generally provided with your ISP subscription. Note: the modem does not currently offer the possibility to get this address during PPP negotiation. AT#DNSSERV1
Secondary DNS	The IP address of the secondary (backup) DNS server must be stored in dotted decimal notation (xxx.xxx.xxx.xxx). This parameter is optional for running the DNS service but provides better reliability of this service (providing a secondary source of DNS resolution if the primary source fails for some reason). This address is generally provided with your ISP subscription. The modem does not currently offer the possibility to get this address during PPP negotiation. AT#DNSSERV2

Operation:

Error codes	Whenever the modem is unable to reach the primary DNS server or when the primary DNS server informs the modem that it can't resolve a symbolic hostname into an IP address, the modem then automatically tries to contact the secondary DNS server. Whenever the modem can't reach the secondary DNS server or when the secondary DNS server informs the modem that it can't resolve a symbolic name into an IP address, a '#CME ERROR: 38027' message is sent on the serial port.

Note:

In the case of GPRS, the DNS server addresses are provided by the GPRS operator on successful network connection of the module. The addresses provided are used in place of DNSSERV1 and DNSSERV2. (The DNS address parameters are not updated for display purposes). If no addresses are provided, the DNSSERV1 and DNSSERV2 parameter addresses are used as in GSM.

Connecting to an Internet Service Provider (ISP) through GSM Data

This example shows how to initiate a GSM data communication from the modem connecting to an ISP.

Description: The attached equipment can initiate an Internet connection, via the modem configured in GSM mode (GPRSMODE parameter), by sending the 'AT#CONNECTIONSTART' command over the serial port.

Upon receiving this command, the modem automatically attempts to complete a PPP connection session according to the relevant stored parameters. This is achieved by the modem which dials the required ISP phone number as predefined in the DIALSELECT parameter.

Once the PPP link is established, the software sends the ISP-assigned dynamic IP address and the 'Ok_Info_Ppp' message as a response code to the attached equipment via the serial port. The modem will then wait for commands from the attached equipment.

Syntax: AT#CONNECTIONSTART

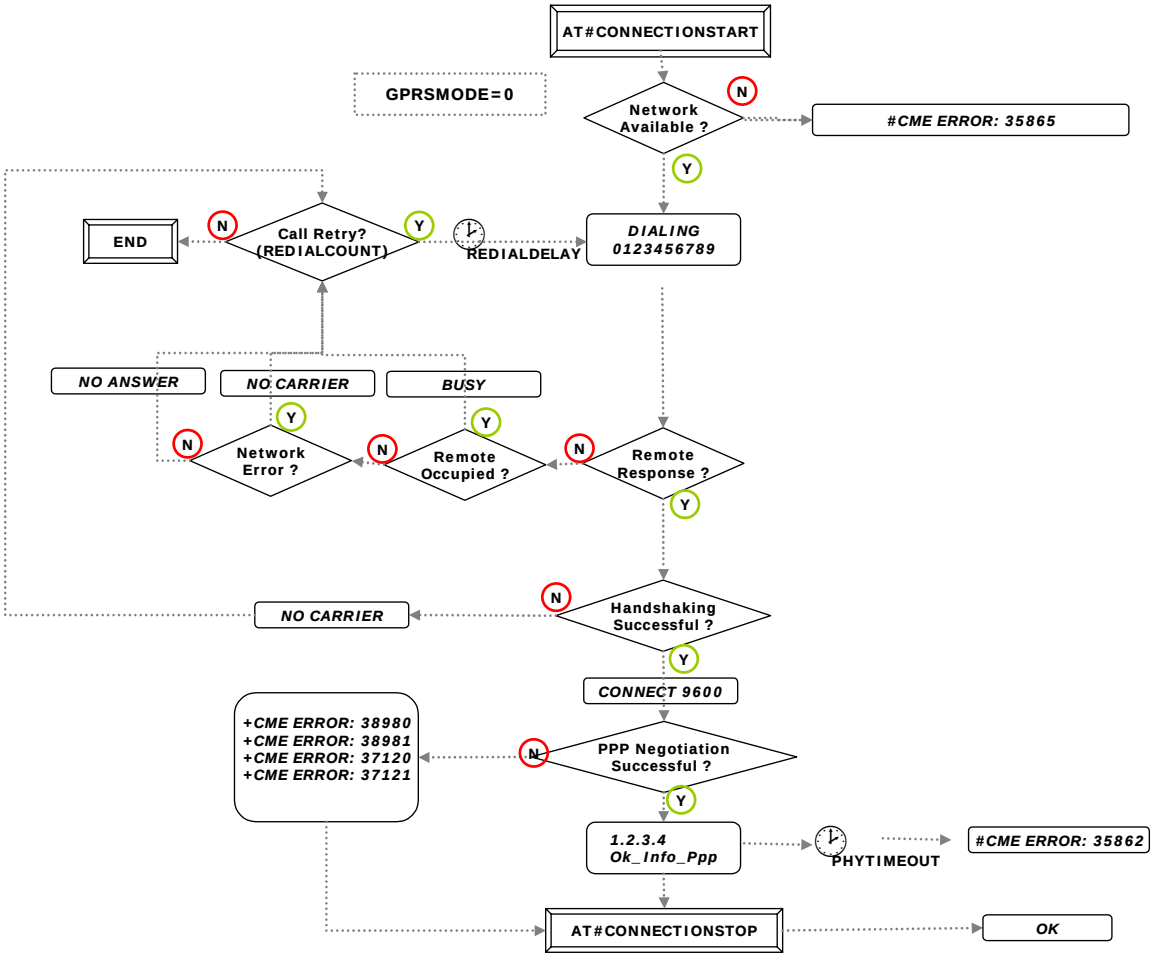
Related Parameters:

Registration on the network AT+ command	The product has to be registered on the GSM network AT+ CREG=1
GSM mode	Deactivate the GPRS mode to switch in the GSM mode. AT#GPRSMODE=0
ISP Primary Number	Primary dial-up phone number to connect with the required local ISP. AT#DIALN1
ISP Secondary Number	Secondary dial-up phone number to connect with the required local ISP. AT#DIALN2
Dial Option	The value of this parameter determines which of the two numbers detailed above (i.e. DIALN1 or DIALN2) is used to establish an Internet connection: the primary dial-up number or the secondary dial-up number. AT#DIALSELECT
ISP User Name	User name / login of the ISP account. AT#ISPUN
ISP User Password	Password for ISP account. AT#ISPPW
Number of call retries	Indicates how many unsuccessful connection attempts the software will make before terminating the connection attempt activity. AT#REDIALCOUNT
Delay between call retries	Controls the delay (in seconds), if any, that will be introduced between each call retry if the call is not successful. AT#REDIALDELAY
Stop the AT#CONNECTION START process	The 'AT#CONNECTIONSTART' process can be stopped by sending the 'AT#CONNECTIONSTOP' command over the serial port AT#CONNECTIONSTOP

Connecting to an Internet Service Provider (ISP) through GSM Data Continued

Parameters: PPPMODE – PHYTIMEOUT
Commands: CONNECTIONSTOP

Process diagram: PPP Connection for an outgoing call



Using the TCP/IP Stack as a PPP Server in GSM Mode

This example shows how to authenticate a remote PPP client and assign it an IP address through a GSM call.

Description: A user may want to use a standard PPP client (like MS Dial-up Networking) to access the modem directly. In this case, the user must configure his or her computer (equipped with a modem) with the phone number related to the device, instead of the one of a standard ISP.
Once authenticated, a direct PPP link is opened between the modem and the remote computer.

Notes:

- Active feature in GSM and not in GPRS mode.
- The embedded PPP server allows the PPP negotiation in the PAP mode only (different from the PPP client mode with the PAP or CHAP negotiation).

Configuration:

PPP server activation	When this parameter is set to 1, the modem behaves as a PPP server when an incoming call is received. For specific requirements, it is also possible to configure the modem as a PPP Server only (4) or as PPP server when initiating a call (2). AT#PPPMODE
PPP server authentication	Defines the login / password required to access the modem internal PPP server. The user must specify these values when creating the PPP client on his computer (like for a standard ISP). AT#PPPSERVUN, PPPSERVPW
PPP server IP address	Once the negotiation is successfully completed, this IP address will be the one assigned to the modem internal PPP server. This IP address must be on the same IP network as the PPPPEERIP. AT#PPPMYIP
PPP client IP address	Once the negotiation is successfully completed, this IP address will be the one assigned to the remote PPP client. This IP address must be on the same IP network as the PPPMYIP AT#PPPPEERIP

Operation:

Negotiation	Once the call is accepted and the modem modulation negotiation is successful, the modem automatically behaves as a PPP server. The 'Ok_Info_Ppp' message indicates that the IP link is up. A '#CME ERROR: 37121 or '#CME ERROR: 38981 message (or a 'NO CARRIER') means that the PPP negotiation has failed. This can be due to: • Incorrect authentication parameters • The PPP client doesn't accept the IP address provided by the modem. • The PPP client doesn't support the basic set of PPP options managed by the modem.

Troubleshooting:

PPP Error	On the PPP entry on the remote PC the following points must be checked: • Valid username, password • The IP address must be attributed by the remote PPP server • LCP options must be disabled into the dial up configuration

Other AT# Parameters: PPPMODE – PPPMYIP – PPPPEERIP – PPPSERVPW – PPPSERVUN - ANSWERMODE

Automatic Call-Back to Incoming Calls

This example covers the following: When an incoming call is received, the modem validates the associated CLI (Calling Line Identity), ignores (doesn't answer) the call and then automatically dials a pre-defined phone number.

Description: When receiving an incoming call it is possible for the attached equipment to either accept the call, let the modem answer the call automatically, or to not answer the call and then let the modem automatically call a pre-defined number using the 'call-back mechanism'.
This application note addresses the 'automatic call-back' mechanism implemented in the modem.

Configuration:

Configuration	• RINGCOUNT and ANSWERMODE must be configured in accordance with the ATS0 configuration for not interacting. • The CALLSCREENNUM parameter configuration with a decimal phone number needs the caller line identity service from the operator and the AT+CLIP=1 command sent to the software. AT#RINGCOUNT
Automatic call-back activation	When this parameter is set to 2 or 3, the modem will not answer an incoming call, but instead automatically : • Calls back either of the DIALN1 or DIALN2 numbers (depending on the DIALSELECT parameter setting) or attempts a GPRS connection (depending on the GPRSMODE parameter setting) if ANSWERMODE is set to 2 by internally issuing an AT#CONNECTIONSTART command-- if the associated CLI (Calling Line Identity) matches the CALLSCREENNUM parameter. • Calls back the caller number (CALLSCREENNUM number) if ANSWERMODE is set to 3 by internally issuing an AT#CONNECTIONSTART command-- if the associated CLI (Calling Line Identity) matches the CALLSCREENNUM parameter. AT#ANSWERMODE = *2 → call back the ISP or attempt GPRS connection.* AT#ANSWERMODE = *3 → call back the caller (dependant on the CALLSCREENNUM number parameter setting).*
Caller ID validation	When the GSM line subscription includes the caller ID service, it is possible to direct the modem to validate the CLI (Calling Line Identity) of the incoming call before executing the 'call back' mechanism. Please take care of its format, communicated by the network. The default value = 0 deactivate the mechanism. If no check is necessary the CALLSCREENNUM parameter must be set to '*'. AT#CALLSCREENNUM
Timer before call-back	When the '+CRING' signals have ceased on the GSM line, it is possible to configure the time (in seconds) that the modem will wait before going off hook and dialing the DIALN1 or DIALN2 phone number, or attempt to make a GPRS connection. AT#CALLBACKTIMER

Operation:

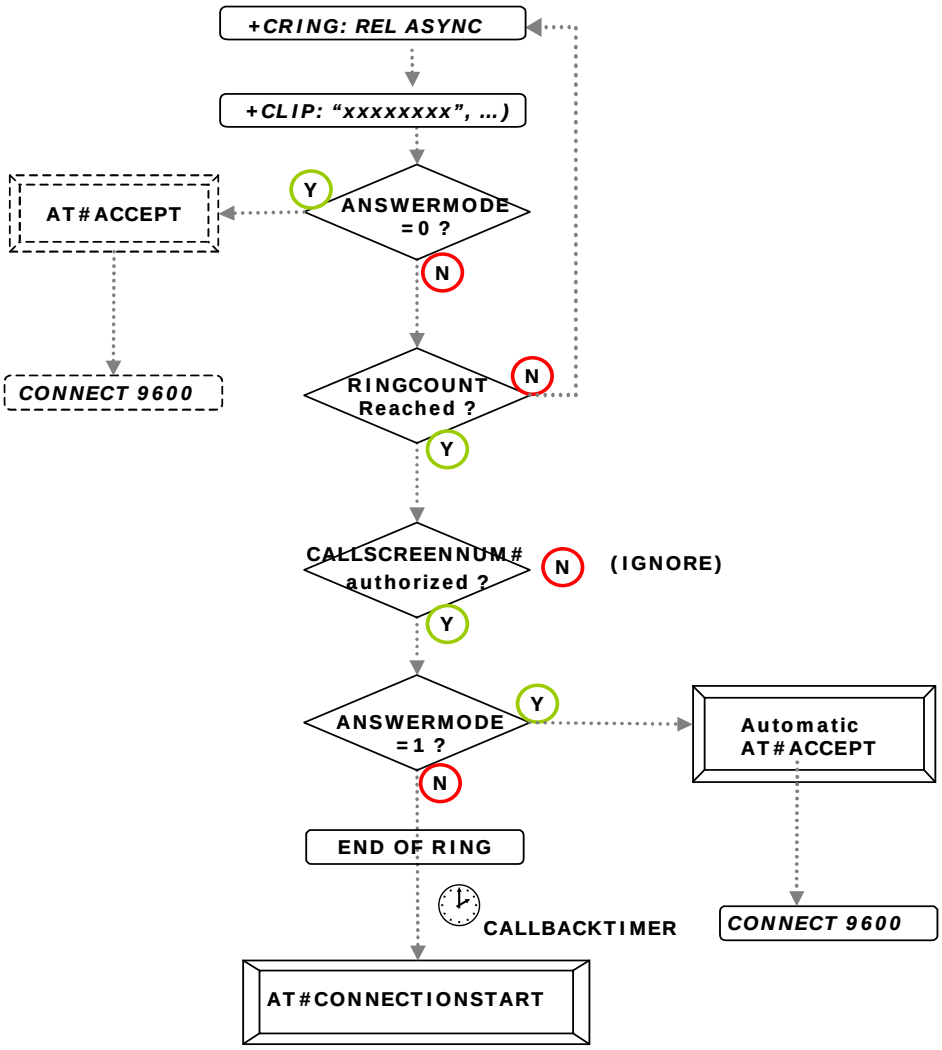
Validation of CLI before call back	When receiving an incoming call, the modem ignores the call but still validates the associated CLI (Calling Line Identity) against the CALLSCREENNUM parameter. If the associated CLI (Calling Line Identity) and CALLSCREENNUM don't match, no further operations take place. If the associated CLI (Calling Line Identity) matches the CALLSCREENNUM parameter, the modem waits for the last RING signal and then for the CALLBACKTIMER period before going off hook and initiating the call back mechanism, by internally issuing the AT#CONNECTIONSTART command. The process is then similar to the standard AT#CONNECTIONSTART process, using GPRSMODE.

Other AT Parameters: AT#PPPMODE, GPRSMODE

Other AT# Commands: AT#CONNECTIONSTART

Automatic Call-Back to Incoming Calls Continued

Process diagram



Automatic Answer of Incoming Calls

This example cover the following: When an incoming call is received, the modem validates the associated CLI (Calling Line Identity) and if appropriately configured, can answer the call automatically.

Description: When receiving an incoming call it is possible for the attached equipment to either accept the call, let the modem answer the call automatically, or to not answer the call and then let the modem automatically call a pre-defined number using the 'call-back mechanism'.

This application note addresses the automatic answer mechanism implemented in the modem.

Configuration:

Configuration	• The value of the PPPMODE parameter will determine the communication type for outgoing and incoming call: IP GSM/GPRS modem, transparent modem, … • The GPRSMODE parameter allows to select the GPRS or GSM mode for outgoing connection to the Internet • The CALLSCREENNUM parameter configuration with a decimal phone number need the caller line identity service from the operator and the AT+CLIP=1 command sent to the software • RINGCOUNT and ANSWERMODE must be configured in accordance with the ATS0 configuration for not interacting.
Automatic answer activation	When this parameter is set to 1, the modem automatically answers an incoming call if the associated CLI (Calling Line Identity) matches the CALLSCREENNUM parameter. AT#ANSWERMODE=1
Caller ID validation	When the GSM line subscription includes the caller ID service, it is possible to direct the modem to validate the CLI (Calling Line Identity) of the incoming call before answering. If no check is required, the CALLSCREENNUM parameter should be set to '*'. AT#CALLSCREENNUM
Number of rings	The configurable number of incoming rings before the modem goes off hook to answer the call. AT#RINGCOUNT

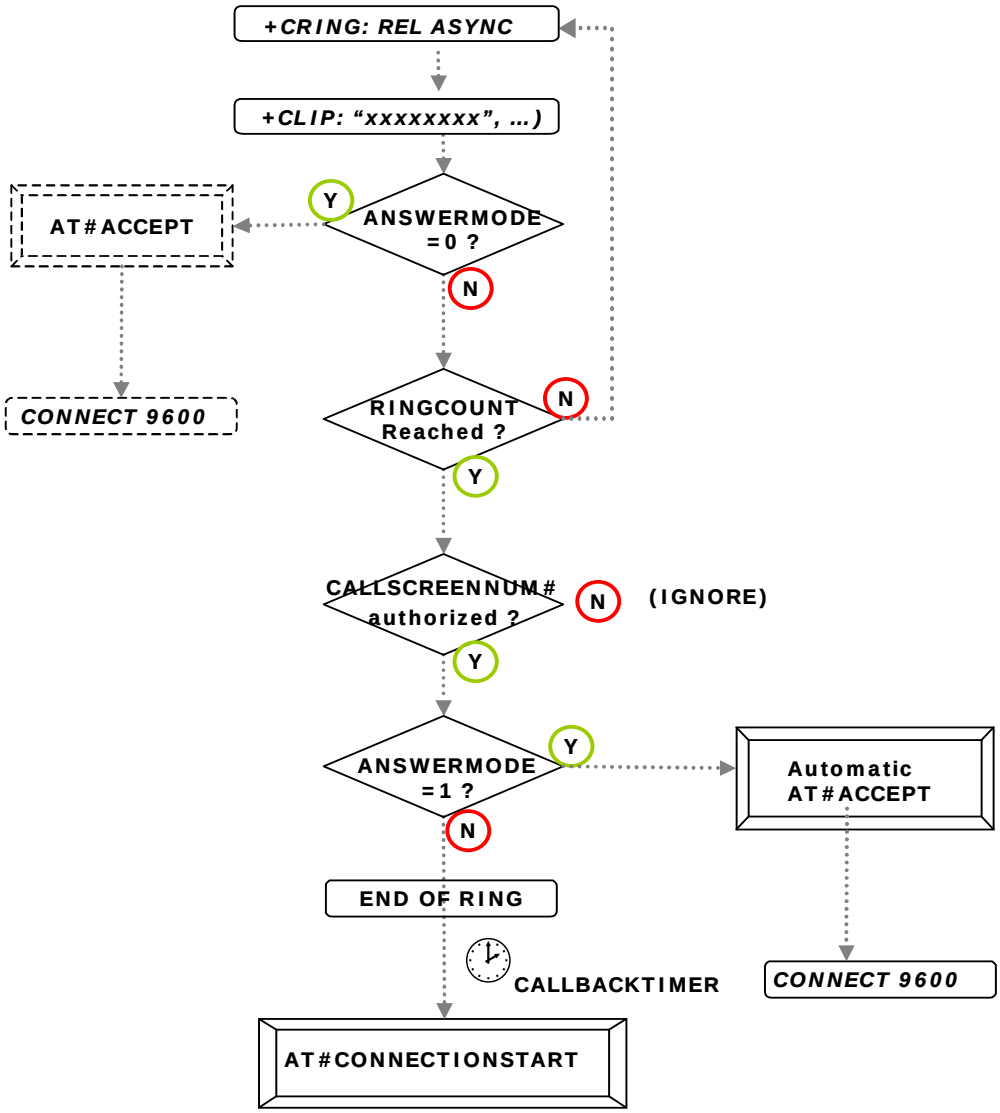
Operation:

Answering	When receiving an incoming call, the modem ignores the call if the associated CLI (Calling Line Identity) doesn't match the CALLSCREENNUM parameter. If the associated CLI (Calling Line Identity) matches the CALLSCREENNUM parameter, the modem goes off hook and tries to establish a connection as specified by the PPPMODE parameter (PPP client, PPP server or transparent modem).

Other AT Parameters: AT#PPPMODE - GPRSMODE

Automatic Answer of Incoming Calls Continued

Process diagram



Answering an Incoming Call

This example cover the following: When an incoming call is received, the modem informs the attached equipment of the call. The attached equipment then decides if the call should be accepted.

Description: When receiving an incoming call it is possible for the attached equipment to either accept the call, to let the modem answer the call automatically, or to not answer the call and then to let the modem automatically call a pre-defined number using the 'call-back mechanism'.

This application note describes how the attached equipment can instruct the modem to answer an incoming call.

Configuration:

Configuration	• The value of the PPPMODE parameter will determine the communication type: IP modem, transparent modem, • ANSWERMODE must be configured in accordance with the ATS0 configuration for not interacting. • RINGCOUNT must be configured for the number of rings before answering the call. • The CALLSCREENNUM parameter configuration with a decimal phone number need the caller line identity service from the operator and the AT+CLIP=1 command sent to the software
Automatic answer deactivation	When the ANSWERMODE parameter is set to 0, the modem lets the attached host decide whether the incoming call should be accepted or ignored. AT#ANSWERMODE=0

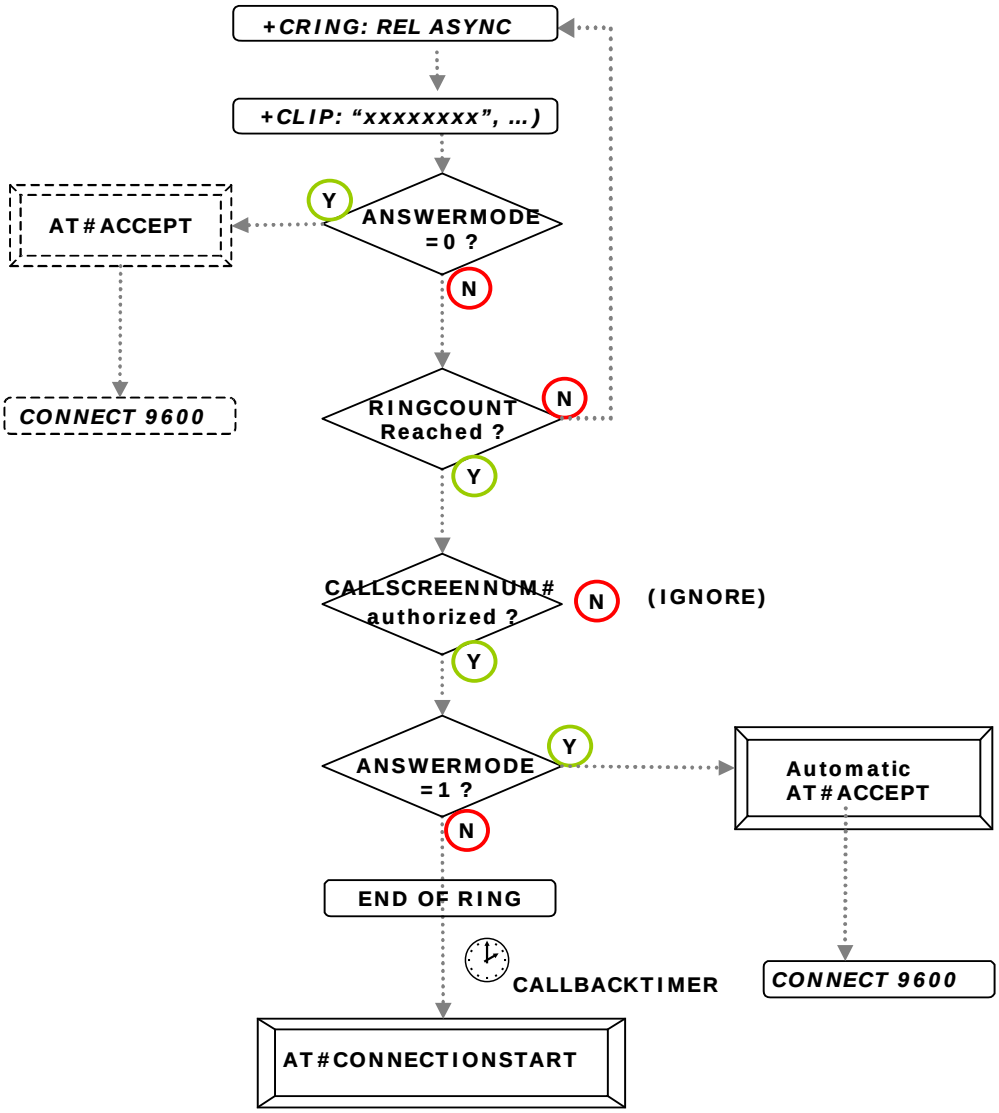
Operation:

Ring information	When receiving an incoming call, the modem informs the attached equipment via the serial port. The modem issues a '+CRING:xxx' message on the serial port for each ring signal received on the GSM line. +CRING
Caller ID information	If the attached network subscription includes a caller ID service, the CLI (Calling Line Identity) is understood by the software. Depending on the telecom carrier, this service may be optional at an additional charge to the subscriber. +CLIP:"xxxxxxxxxx",
Accepting the call	To direct the modem to accept this call, the attached equipment must send the 'AT#ACCEPT' command via the serial port. The modem then goes off hook and attempts to establish the connection as specified in the PPPMODE parameter (PPP client, server or transparent modem). AT# command: AT#ACCEPT
Ignoring the call	If the attached equipment decides that the incoming call should not be answered, it should not send the 'AT#ACCEPT' command via the serial port, the modem will, therefore, ignore the call.

Other AT# Parameters: PPPMODE

Other AT# Commands: CONNECTIONSTART

Process diagram



Connecting to the Internet through GPRS

This example covers the following: Initiate a GPRS communication from the modem for an Internet connection.

Description: The attached equipment can initiate an Internet connection, via the modem configured in GPRS mode (GPRSMODE parameter), by sending the '**AT#CONNECTIONSTART**' command over the serial port.

Upon receiving this command, the modem automatically attempts to complete a GPRS connection session according to the relevant stored parameters. This is achieved by the modem which connects to the Access Point Name Server.

Once the GPRS link is established, the software sends the ISP-assigned dynamic IP address and the 'Ok_Info_GprsActivation' message as a response code to the attached equipment via the serial port. The modem will then wait for commands from the attached equipment.

AT# command: AT#CONNECTIONSTART

Related Parameters:

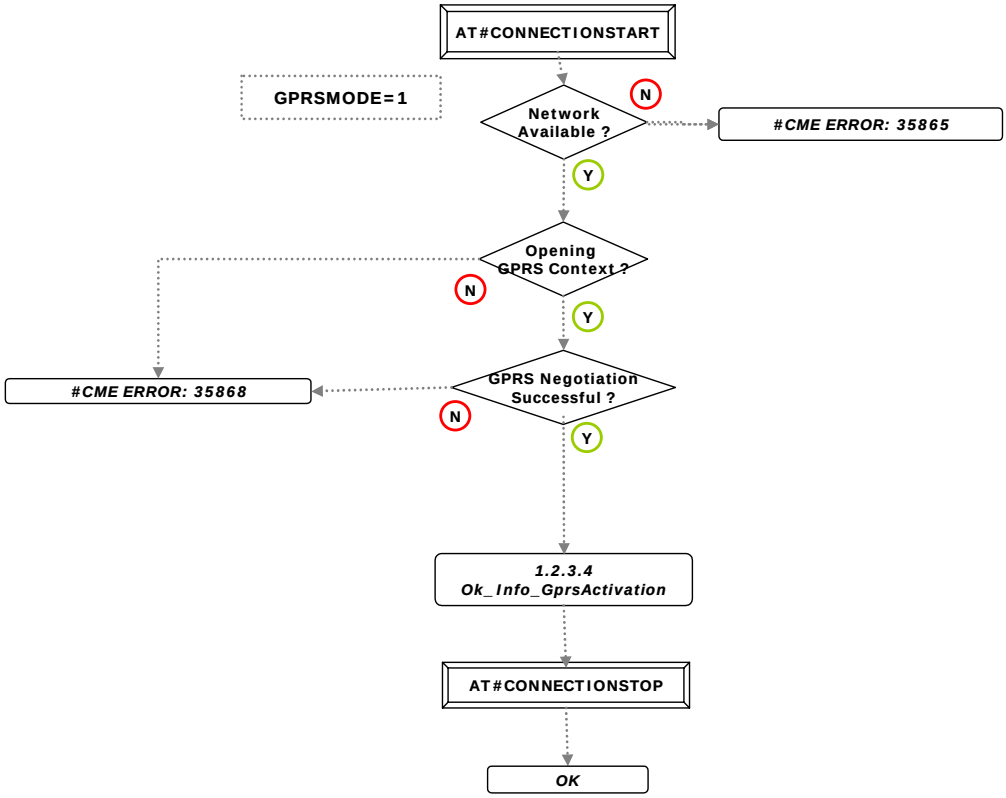
Registration on the network	The product has to be attached and registered on the GPRS network (and on GSM network if needed for the application). AT+ command: AT+CGATT=1 and then AT+CGREG=1 AT+CREG=1 for GSM
GPRS mode	Activate the GPRS mode for connection. AT#GPRSMODE=1
APN Server	Address of the Access Point Name Server required by the network operator for connecting to the Internet. AT#APNSERV
APN User Name	Username required for the connection to the APN. AT#APNUN
APN Password	Password coming with the Username for connecting to the APN. AT#APNPW
Stop the AT#CONNECTION START process	The 'AT#CONNECTIONSTART' process can be stopped by sending the 'AT#CONNECTIONSTOP' command over the serial port AT#CONNECTIONSTOP

Other AT# Parameters: PPPMODE - GPRSMODE

Other AT# Commands: CONNECTIONSTOP

Connecting to the Internet through GPRS Continued

Process diagram: GPRS Connection for an outgoing call



Retrieving Email

This example covers the following: Retrieving email from a POP3 account.

Description: This command allows another processor (attached equipment) to instruct the modem to retrieve an email message. While the modem is connected to the Internet, the attached equipment can retrieve an email message at any time (except when the modem is already in the process of sending or retrieving email, in an FTP process or socket TCP process). The 'AT#GETMAIL' command sent over the serial port instructs the modem to retrieve one email per retrieve command.

Retrieving email	• The modem sends an 'Ok_Info_Mail' message, before sending the email over the serial port if any are present. This message notifies the host to switch from command mode to data mode. At the end of the data flow, the sequence [CR][LF]. [CR][LF] notifies the host to switch from data mode to command mode. • - When there is no mail waiting in the POP3 account, the message 'Ok_Info_NoMail' is sent by the modem over the serial port. AT# Command: AT#GETMAIL **Notes:** • The communication session must be established (AT#CONNECTIONSTART) before sending the 'AT#GETMAIL' command. After the retrieval process, the modem keeps the communication established. • When the email is retrieved, it is deleted from the POP3 server.

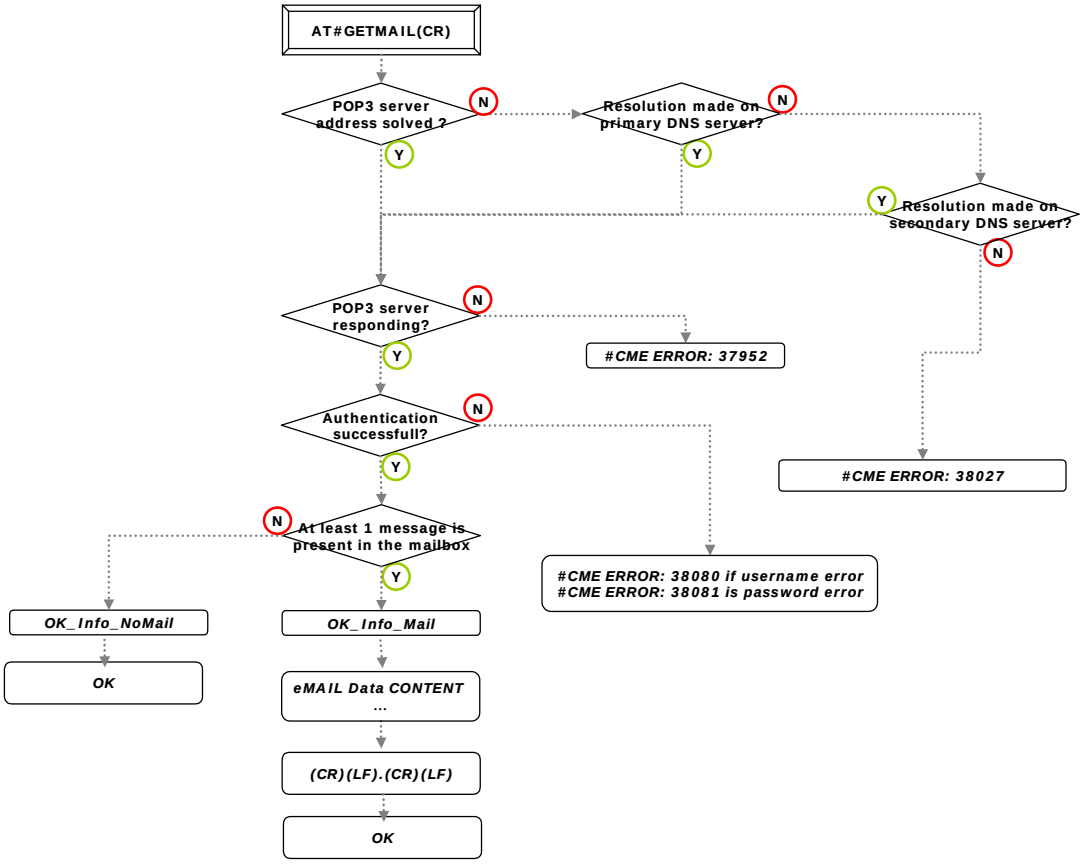
Configuration:

POP3 server	Address of the POP3 server used for retrieving email. AT#POP3SERV
POP3 port	Port of the POP3 server used for retrieving email. AT#POP3PORT
POP3 user name	POP3 user name used for account identification. AT#POP3UN
POP3 user password	POP3 user password used for user authentication. AT#POP3PW
POP3 header	Configuration for sending (1) or not (0) the email header over the serial port while retrieving the email message. AT#POP3HEADERMODE

Other AT Commands: DNSSERV1, DNSSERV2

Retrieving Email Continued

Process diagram: Email retrieving



Sending an Email Containing Data from the Serial Port

This example covers the following: Sending an email message containing data from the serial port.

Description: This command allows another processor (attached equipment) to send an email message containing body text which is passed directly to the modem over the serial port. While the modem is connected to the Internet, the attached equipment can initiate the sending of an email message at any time (except when the software is already in the process of sending or retrieving email, FTP or socket TCP process). This action is accomplished by sending an `'AT#PUTMAIL'` command over the serial port.

Sending an email through the serial port	The modem will send an 'Ok_Info_WaitingForData' message when it is ready to accept the data flow to be placed in the body of the email from the serial port. The data sent by the host over the serial port is not echoed. **Note:** When using this command, the software will use the following parameters to compose the email header: 'SUBJ1' for the email subject 'REC1' for the recipient email address 'CCREC1' for the carbon Copy recipient -mail address 'SENDERNAME' & 'SENDERADDR' for the specified identity of the module **AT# Command: AT#PUTMAIL** The end of the email body data flow must be denoted by sending following sequence to the module: [CR][LF].[CR][LF] The software will send an 'OK' response when it is ready to accept another command, in case of success of the sending operation. In case of problem a '#CME ERROR: Value' is sent over the serial port. **Notes:** • The [CR][LF].[CR][LF] keyboard sequence is: ENTER CTRL+ENTER . ENTER CTRL+ENTER • If there is no answer from the SMTP server after 3 minutes, the error #CME ERROR: 37964 is generated and the sending process is aborted. • A communication session must be established (AT#CONNECTIONSTART) before sending the AT#PUTMAIL command. After the sending process, the communication session remains established. • Check your terminal emulator configuration because with some configurations, this terminal can modify data (for example, to transform the [CR] in [CR][LF]).

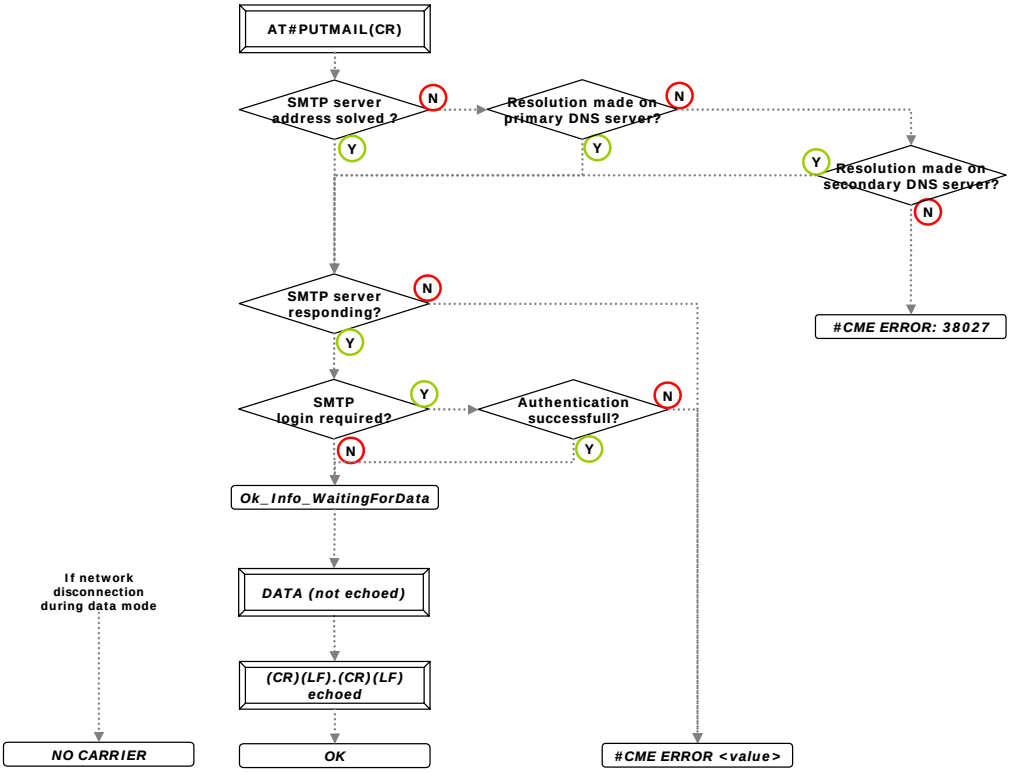
Configuration:

SMTP server	Address of the SMTP server used for sending email. AT#SMTPSERV
SMTP port	Port of the SMTP server used to send email. AT#SMTPPORT
SMTP user name	SMTP user name used by some SMTP servers for authentication. AT#SMTPUN
SMTP user password	SMTP user password used by some SMTP servers for authentication. AT#SMTPPW
Domain	Domain name of the sender. AT#DOMAIN
Sender name	The sender's name, which appears in the email header, in the field: 'From:' AT#SENDERNAME
Sender email address	Email address of the sender, which appears in the email header, in the field ' From:' AT#SENDERADDR
Recipient email address	Email address of the recipient, which appears in the email header, in the field ' To:' AT#REC1
CC Recipient email address	Carbon Copy email address of the recipient for sending email which appears in the email header, in the field ' Cc:' AT#CCREC1
Email subject	Pre-defined email subject, which appears in the email header, in the field ' Subject:' AT#SUBJ1

Sending an Email Containing Data from the Serial Port Continued

Other AT# Commands: SENDMAIL1 – SENDMAIL2 – SENDMAIL3

Process diagram



Notes:	• The data in the email sent will not be echoed back to the host. • - The <CR><LF>.<CR><LF> sequence denoting the end of data mode is echoed back.

Sending a Pre-Defined Email Combination

This example covers the following: Sending an email composed by a pre-defined combination of parameters stored into the Flash memory.

Description: This command sends one of 3 pre-defined email combinations. This command will assemble the parameters stored in the flash memory to compose and send the relevant email message combination.

Sending a pre-defined email	The 'AT#SENDMAIL1' (or AT#SENDMAIL2, AT#SENDMAIL3) command will compose the email using the following parameters: 'SUBJ1' (or SUBJ2, SUBJ3) for the email subject 'REC1' (or REC2, REC3) for the recipient email address 'CCREC1' (or CCREC2, CCREC3) for the carbon copy recipient email address 'BODY1' (or BODY2, BODY3) for the email body which has a limited size of 120 characters 'SENDERNAME' & 'SENDERADDR' for the specific identity of the module The response message is only 'OK' when the process is successful. The body content is not echoed on the serial port. **AT# Command:** AT#SENDMAIL1 / AT#SENDMAIL2 / AT#SENDMAIL3 **Notes:** • The communication session must be established (AT#CONNECTIONSTART) before sending this command. After the sending process, the modem will keep the communication established. • If the RECn, SENDERADDR and DOMAIN parameters are empty, the response from the modem to the AT#SENDMAILn command will be an associated error code.

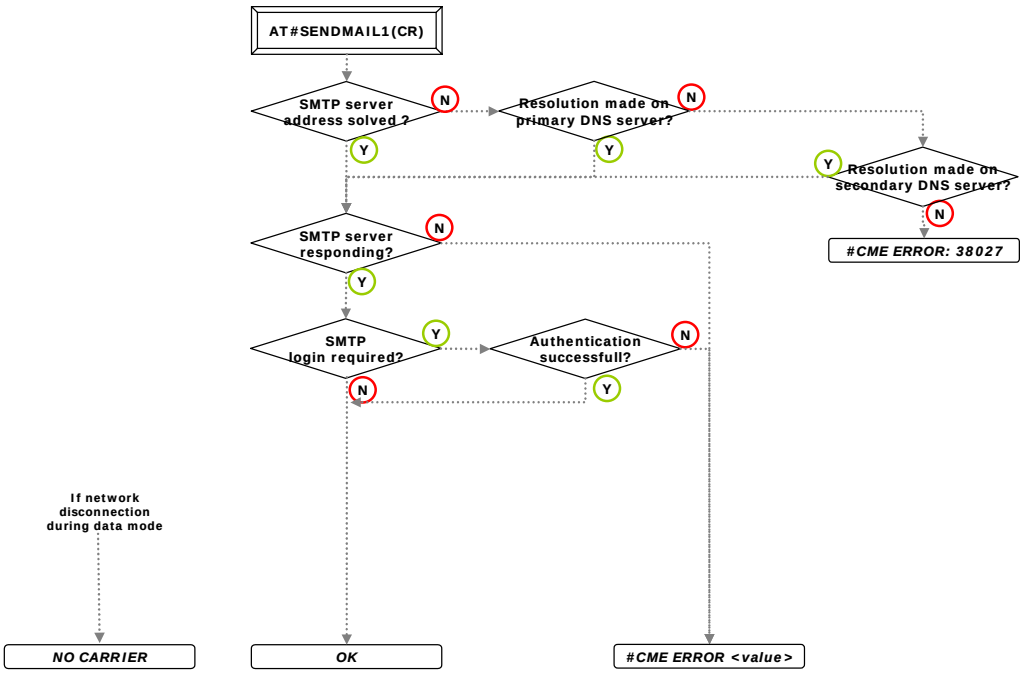
Other AT# Commands: AT#PUTMAIL

Configuration:

SMTP server	Address of the SMTP server used for sending emails. AT#SMTPSERV
SMTP port	Port of the SMTP server used for the email sending. AT#SMTPPORT
SMTP user name	SMTP user name used by some SMTP servers for authentication. AT#SMTPUN
SMTP user password	SMTP user password used by some SMTP servers for authentication. AT#SMTPPW
Domain	Domain name of the sender. AT#DOMAIN
Sender name	The sender's name which appears in the email header, in the field: 'From:' AT#SENDERNAME
Sender email address	Email address of the sender which appears in the email header, in the field ' From:' AT#SENDERADDR
Recipient email address	Email address of the recipient for sending email which appears in the email header, in the field ' To:' AT#REC1 – REC2 – REC3
CC Recipient email address	Carbon Copy email address of the recipient for sending email which appears in the email header, in the field ' Cc:' AT#CCREC1 – CCREC2 – CCREC3
Email subject	Pre-defined email subject which appears in the email header, in the field ' Subject:' AT#SUBJ1 – SUBJ2 – SUBJ3
Email Body	Pre-defined email body AT#BODY1 – BODY2 – BODY3

Sending a Pre-Defined Email Combination Continued

Process diagram



Note	The data into the email sent will not be echoed back, the software transparently switches from command to data mode then back to command mode

Downloading an FTP File

This example covers the following: Downloading a file from a remote FTP server.

Configuration:

FTP server address	Defines the address of the remote FTP server. AT#FTPSERV
FTP client authentication	Defines the login / password required to access the FTP server. AT#FTPUN, FTPPW
FTP file path and name	Prior to instructing the modem to retrieve a file from a FTP server, it is necessary to specify the path and the name of the relevant file to download. AT#FTPGETPATH, FTPGETFILENAME
FTP server ports	For some installations (due to specific firewall, proxy or router configurations) the TCP control port number used by the remote FTP server may need to be changed. In most of the cases this is not necessary and the default value for the FTP control port is 21. AT#FTPPORT

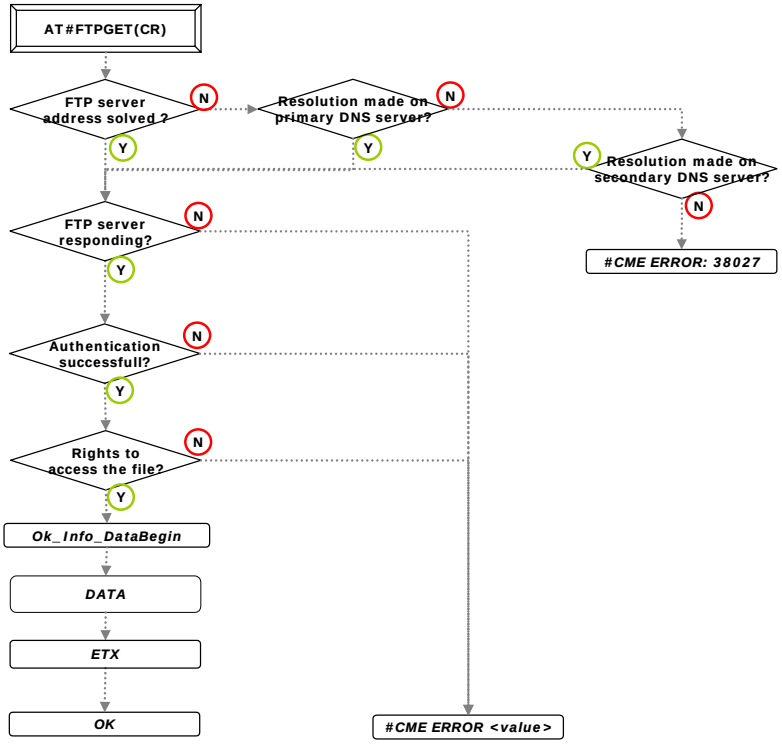
Operation:

Launching the download	Once an IP link is established, this command instructs the modem to logon to the remote FTP server and then to start the download of the specified file by issuing a standard GET command. AT#FTPGET
Data flow	The modem response code 'Ok_Info_DataBegin' indicates to the attached equipment that the data connection is about to be opened for transmitting the file. Then, all of the file data received from the connection is sent over the serial port without echo. The [ETX] character is sent by the modem over the serial port to indicate that the data connection is closed (meaning the file transfer is completed or has been interrupted). As a consequence an [ETX] character received in the TCP payload data will be transmitted on the serial port as a [DLE][ETX] and a [DLE] as a [DLE][DLE]. It is the attached host's responsibility to decode the received data.
End of download	The modem indicates the closing of the data connection by sending an [ETX] character which is echoed. The response message that follows the data transfer, sent by the modem, signals whether the file has been successfully received or not.

Other AT# Parameters: FTPSERV, FTPUN, FTPPW, FTPGETPATH, FTPGETFILENAME, FTPPORT
Other AT# Commands: FTPGET

Downloading an FTP File Continued

Process diagram



Uploading an FTP File

This example covers the following: Uploading a file to a remote FTP server.

Configuration:

FTP server address	Defines the address of the remote FTP server. AT#FTPSERV
FTP client authentication	Defines the login / password required to access the FTP server. AT#FTPUN, FTPPW
FTP file path and name	Prior to instructing the modem to put a file onto the FTP server, it is necessary to specify the path and the name of the file that will be created on the remote server. AT#FTPPUTPATH, FTPPUTFILENAME
FTP file type	This parameter defines if the file is sent in ASCII (A) or binary mode (I). When set to ASCII all the data sent by the modem to the FTP server is made of 7 bits characters (NVT-ASCII: the MSB is set to 0). As a consequence binary data containing 8 bits characters will be corrupted during the transfer if the FTPTYPE is set to ASCII. AT#FTPTYPE
FTP server ports	For some installations (due to specific firewall, proxy or router configurations) the TCP control port number used by the remote FTP server may need to be changed. In most of the cases this is not necessary and the default value for the FTP control port is 21. AT#FTPPORT

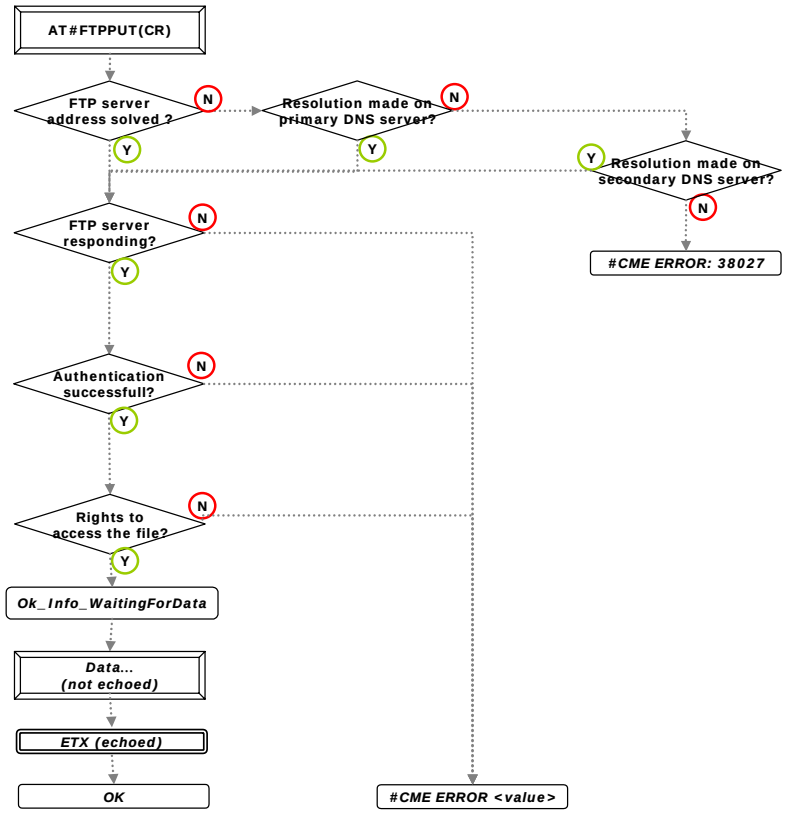
Operation:

Starting the upload	Once an IP link is established, this command directs the modem to logon to the remote FTP server and then to start the upload of the specified file by issuing a standard PUT command. AT#FTPPUT
Data flow	The modem returns the 'Ok_Info_WaitingForData' message for indicating to the attached host that the data connection has been opened for transmitting the file. Then, all the data received from the serial port is sent during the data connection session without echo. The [ETX] character is sent by the attached host and echoed to notify that the data connection is closed (meaning the host has completed the file transfer). As a consequence, an [ETX] character to be sent within the file must be transmitted on the serial port as a [DLE][ETX] and a [DLE] as a [DLE][DLE]. The modem then decodes the data received on the serial port and sends [ETX] or [DLE] in the payload data.
Stopping the upload	The attached host can stop the transfer by sending an [ETX] character. This character is echoed.
Stopping the upload	The attached host can stop the transfer by sending an [ETX] character. This character is echoed. [ETX] CTRL+C sequence in a keyboard.
End of upload	The host indicates to the modem the closing of the data connection by sending an [ETX] character which is echoed. The response message that follows the data transfer, sent by the modem, signal whether the file has been successfully received or not : 'OK' or '#CME ERROR: Value'

Note: The user rights for over-writing an existing file in a FTP server depends on the FTP server's configuration and if incorrectly set can result in failure of the FTP attempt.

Other AT# Parameters: FTPSERV, FTPUN, FTPPW, FTPPUTPATH, FTPPUTFILENAME, FTPTYPE, FTPPORT
Other AT# Commands: FTPPUT

Process diagram



Setting a TCP Port to a Listening Mode

This example covers the following: Opening locally and listen to a TCP socket to accept incoming connections from remote TCP clients.

Description: The modem provides a TCP socket API that allows the user to open and listen to a TCP connection on a configurable TCP port. The IP address of the remote client can be filtered if necessary.

Only one TCP session can be opened simultaneously.

Once the TCP session is opened, the modem acts as a transparent bridge between the TCP connection and the serial connection.

For a GPRS Internet connection, make sure that the operators are allowed to use the TCP port used by your application.

Configuration:

TCP client address	Defines the IP address of the remote TCP client allowed to open the socket TCP. It must be configured as a 32-bit number in dotted-decimal notation - not a symbolic name. When this parameter is set to 255.255.255.255, the access is granted for all remote IP addresses. AT#TCPSERV
TCP server port	Defines the TCP port number of the listening TCP socket. It can be configured with a 5-digit number. For example, the TCP port assigned to Telnet is 23. It must be the same between the peers. AT#TCPPORT

Operation:

Launching the Listening TCP session	Once an IP link is established, this command instructs the modem to listen to the incoming connections on the TCP port specified in the TCPPORT parameter and issued by the allowed IP address defined in the TCPSERV parameter. AT#LTCPSTART
Opening the TCP session	Upon detection of an incoming TCP connection from an allowed IP address on the specified TCP port, the modem automatically opens the TCP session and issues an 'Ok_Info_WaitingForData' message over the serial port.
Data flow	Once the TCP socket has been successfully opened, all the data received from this session is sent over the serial port and all the data received on the serial port is sent to the remote host within TCP packets. Data received from the host are not echoed. This link is bi-directional and the flow is controlled through the selected flow control mechanism. If DLEMODE is set to 1, the attached host must send the [ETX] character as [DLE] [ETX] and the [DLE] as [DLE] [DLE]. On the other hand, an [ETX] character received in the TCP payload data will be transmitted over the serial port as a [DLE] [ETX] and a [DLE] as a [DLE] [DLE]. AT#DLEMODE
Closing the TCP session from the remote	• If the remote ends the TCP session- (and if DLEMODE is set to 1)- the modem will also close the socket TCP session by sending an [ETX] character over the serial port to the host, then the message 'Ok_Info_SocketClosed'. • If DLEMODE is set to 1, the attached host sends an [ETX] character to direct the modem to close the current TCP session. In this case the 'Ok_Info_SocketClosed' message signals that the TCP socket has been closed. An 'OK' message then indicates that the TCP process is completely finished. As soon as the socket TCP is closed, the TCP port is not anymore in a listening mode. If DLEMODE is set to 0, it is not possible for the attached host to close the TCP session in data mode. [ETX] CTRL+C sequence in a keyboard [DLE] CTRL+P sequence in a keyboard
Closing the TCP session from the local host	When the modem is in TCP listening mode (launched by the AT#LTCPSTART command), this can be stopped by the AT#LTCPSTOP command. An 'OK' message then informs that the TCP process is completely finished. If DLEMODE is set to 0, it is not possible for the attached host to close the TCP session. AT#LTCPSTOP

Other AT# Parameters: TCPSERV, TCPPORT, DLEMODE

Other AT# Commands: LTCPSTART, LTCPSTOP, OTCP

Opening a TCP Socket

This example covers the following: Opening and using a TCP socket to a remote TCP server.

Description: The modem provides a TCP socket API that allows the user to open a TCP session with a remote host. The user can freely configure the IP address and TCP port of the remote server.

Only one TCP socket can be opened simultaneously.

Once the TCP session is opened, the modem acts as a transparent bridge between the TCP connection and the serial port.

For a GPRS Internet connection, make sure that the operators are allowed to use the TCP port used by your application.

Configuration:

TCP server address	Defines the IP address of the remote TCP server. It must be configured as a 32-bit number in dotted-decimal notation or as a symbolic name. AT#TCPSERV
TCP server port	Defines the TCP port number of the remote TCP server. It can be configured with a 5-digit number. For example the TCP port assigned to Telnet is usually 23. It must be the same between the peers. AT#TCPPORT
Closing mechanism and DLE ETX coding	Once the TCP connection is opened, either side can close it (the module's ability to do this depends on the DLEMODE parameter setting). When the attached equipment wants to close the session, it must send an [ETX] character over the serial port. In this case a coding mechanism (DLE based) must be applied to allow the attached equipment to send [ETX] characters in the payload data. The DLEMODE parameter defines if [ETX] characters received on the serial port must be interpreted by the modem as a close command. When the DLEMODE parameter is set to 0, an open TCP session can only be closed by the remote TCP server. AT#DLEMODE [ETX] CTRL+C sequence in a keyboard [DLE] CTRL+P sequence in a keyboard

Operation:

Opening the TCP session	Once an IP link is established, this command instructs the modem to open a connection to the remote TCP server as specified in the TCPSERV/TCPPORT parameters. An 'Ok_Info_WaitingForData' message means that the operation has succeeded. AT#OTCP
Data flow	Once the TCP socket has been successfully opened, all the data received from this session is sent over the serial port and all the data received on the serial port is sent to the remote host within TCP packets. Data from the host to the remote is not echoed. This link is bi-directional and the flow is controlled through the hardware flow control mechanism. If DLEMODE is set to 1, the attached host must send the [ETX] character as [DLE] [ETX] and the [DLE] as [DLE] [DLE]. On the other hand, an [ETX] character received in the TCP payload data will be transmitted over the serial port as a [DLE] [ETX] and a [DLE] as a [DLE] [DLE].
Closing the TCP session	If DLEMODE is set to 1, the attached host sends an [ETX] character to direct the modem to close the current TCP session. This character is echoed In this case the 'Ok_Info_SocketClosed' message signals that the TCP socket has been closed. An 'OK' message then indicates that the TCP process is completely finished. If DLEMODE is set to 0, it is not possible for the attached host to close the TCP session. [ETX] CTRL+C sequence in a keyboard [DLE] CTRL+P sequence in a keyboard

Other AT# Parameters: DLEMODE, TCPPORT, TCPSERV

Other AT# Commands: LTCPSTART, LTCPSTOP, OTCP